NISTIR 7904

Trusted Geolocation in the Cloud: Proof of Concept Implementation

Michael Bartock
Murugiah Souppaya
Raghuram Yeluri
Uttam Shetty
James Greene
Steve Orrin
Hemma Prafullchandra
John McLeese
Jason Mills
Daniel Carayiannis
Tarik Williams
Karen Scarfone

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.7904

National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 7904

Trusted Geolocation in the Cloud: Proof of Concept Implementation

Michael Bartock
Murugiah Souppaya
Computer Security Division
Information Technology Laboratory

Raghuram Yeluri
Uttam Shetty
James Greene
Steve Orrin
Intel Corporation
Santa Clara, California

Hemma Prafullchandra
John McLeese
Jason Mills
HyTrust
Mountain View, California

Daniel Carayiannis
Tarik Williams
RSA, The Security Division of EMC
Bedford, Massachusetts

Karen Scarfone
Scarfone Cybersecurity
Clifton, Virginia

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.7904

December 2015

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

National Institute of Standards and Technology Internal Report 7904
59 pages (December 2015)

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.IR.7904

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by Federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Comments on this publication may be submitted to:

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems.

Abstract

This publication explains selected security challenges involving Infrastructure as a Service (IaaS) cloud computing technologies and geolocation. It then describes a proof of concept implementation that was designed to address those challenges. The publication provides sufficient details about the proof of concept implementation so that organizations can reproduce it if desired. The publication is intended to be a blueprint or template that can be used by the general security community to validate and implement the described proof of concept implementation.

Keywords

cloud computing; geolocation; Infrastructure as a Service (IaaS); roots of trust; virtualization

Acknowledgments

The authors wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content, in particular Kevin Fiftal from Intel Corporation.

Audience

This document has been created for security researchers, cloud computing practitioners, system integrators, and other parties interested in techniques for solving the security problem in question: improving the security of virtualized infrastructure cloud computing technologies by enforcing geolocation restrictions.

Trademark Information

All registered trademarks or trademarks belong to their respective organizations.

Table of Contents

List of Appendices

List of Figures and Tables

1 Introduction

1.1 Purpose and Scope

This publication explains selected security challenges involving Infrastructure as a Service (IaaS) cloud computing technologies and geolocation. It then describes a proof of concept implementation that was designed to address those challenges. The publication provides sufficient details about the proof of concept implementation so that organizations can reproduce it if desired. The publication is intended to be a blueprint or template that can be used by the general security community to validate and implement the described proof of concept implementation.

It is important to note that the proof of concept implementation presented in this publication is only one possible way to solve the security challenges. It is not intended to preclude the use of other products, services, techniques, etc. that can also solve the problem adequately, nor is it intended to preclude the use of any cloud products or services not specifically mentioned in this publication.

1.2 Document Structure

This document is organized into the following sections and appendices:

- Section 2 defines the problem (usage scenario) to be solved.
- Sections 3, 4, and 5 describe the three stages of the proof of concept implementation.
- Appendix A provides an overview of the high-level hardware architecture of the proof of concept implementation, as well as details on how Intel platforms implement hardware modules and enhanced hardware-based security functions.
- Appendix B contains supplementary information provided by HyTrust describing all the required components and steps required to setup the proof of concept implementation.
- Appendix C contains supplementary information provided by Intel describing all the required components and steps required to setup the proof of concept implementation.
- Appendix D presents screen shots from the HyTrust CloudControl product that demonstrate the monitoring of measurements in a governance, risk, and compliance dashboard.
- Appendix E presents screen shots from the RSA Archer product that demonstrate the monitoring of measurements in a governance, risk, and compliance dashboard.
- Appendix F lists the major controls from NIST Special Publication 800-53 Revision 4, *Security and Privacy Controls for Federal Information Systems and Organizations* that affect trusted geolocation.
- Appendix G maps the major security features from the proof of concept implementation to the corresponding subcategories from the Cybersecurity Framework.
- Appendix H lists and defines acronyms and other abbreviations used in the document.
- Appendix I provides references for the document.

2 Usage Scenario

This section defines the problem—the *usage scenario*—that is to be solved through the proof of concept implementation. Section 2.1 explains the basics of the problem. Section 2.2 defines the problem more formally, outlining all of the intermediate requirements (goals) that must be met in order to achieve the desired solution. These requirements are grouped into three stages of the usage scenario, each of which is examined more closely in Sections 2.2.1 through 2.2.3, respectively.

2.1 Problem to Address

Shared cloud computing technologies are designed to be highly agile and flexible, transparently using whatever resources are available to process workloads for their customers. However, there are security and privacy concerns with allowing unrestricted workload migration. Whenever multiple workloads are present on a single cloud server, there is a need to segregate those workloads from each other so that they do not interfere with each other, gain access to each other's sensitive data, or otherwise compromise the security or privacy of the workloads. Imagine two rival companies with workloads on the same server; each company would want to ensure that the server can be trusted to protect their information from the other company. Similarly, a single organization might have multiple workloads that need to be kept separate because of differing security requirements and needs for each workload.

Another concern with shared cloud computing is that workloads could move from cloud servers located in one country to servers located in another country. Each country has its own laws for data security, privacy, and other aspects of information technology (IT). Because the requirements of these laws may conflict with an organization's policies or mandates (e.g., laws, regulations), an organization may decide that it needs to restrict which cloud servers it uses based on their location. A common desire is to only use cloud servers physically located within the same country as the organization, or physically located in the same country as the origin of the information. Determining the approximate physical location of an object, such as a cloud computing server, is generally known as *geolocation*. Geolocation can be accomplished in many ways, with varying degrees of accuracy, but traditional geolocation methods are not secured and they are enforced through management and operational controls that cannot be automated and scaled. Therefore, traditional geolocation methods cannot be trusted to meet cloud security needs.

The motivation behind this usage scenario is to improve the security of cloud computing and accelerate the adoption of cloud computing technologies by establishing an automated hardware root of trust method for enforcing and monitoring geolocation restrictions for cloud servers. A hardware root of trust is an inherently trusted combination of hardware and firmware that maintains the integrity of the geolocation information and the platform. The hardware root of trust is seeded by the organization, with the host's unique identifier and platform metadata stored in tamper-resistant hardware. This information is accessed by management and security tools using secure protocols to assert the integrity of the platform and confirm the location of the host.

2.2 Requirements

Using trusted compute pools (described in Section 3) is a leading approach to aggregate trusted systems and segregate them from untrusted resources, which results in the separation of higher-value, more sensitive workloads from commodity application and data workloads. The principles of operation are to:

1. Create a part of the cloud to meet the specific and varying security requirements of users.

2. Control access to that portion of the cloud so that the right applications (workloads) get deployed there.

3. Enable audits of that portion of the cloud so that users can verify compliance.

These trusted compute pools allow IT to gain the benefits of the dynamic cloud environment while still enforcing higher levels of protections for their more critical workloads.

The ultimate goal is to be able to use trusted geolocation for deploying and migrating cloud workloads between cloud servers within a cloud. This goal is dependent on smaller prerequisite goals, which can be thought of as requirements that the solution must meet. Because of the number of prerequisites, they have been grouped into three stages:

0. **Platform Attestation and Safer Hypervisor Launch.** This ensures that the cloud workloads are run on trusted server platforms.
1. **Trust-Based Homogeneous Secure Migration.** This stage allows cloud workloads to be migrated among homogeneous trusted server platforms within a cloud.
2. **Trust-Based and Geolocation-Based Homogeneous Secure Migration.** This stage allows cloud workloads to be migrated among homogeneous trusted server platforms within a cloud, taking into consideration geolocation restrictions.

The prerequisite goals for each stage, along with more general information on each stage, are explained below.

2.2.1 Stage 0: Platform Attestation and Safer Hypervisor Launch

A fundamental component of a solution is having some assurance that the platform the workload is running on can be trusted. If the platform is not trustworthy, then not only is it putting the workload at greater risk of compromise, but also there is no assurance that the claimed geolocation of the cloud server is accurate. Having basic assurance of trustworthiness is the initial stage in the solution.

Stage 0 includes the following prerequisite goals:

1. **Configure a cloud server platform as being trusted.** The “cloud server platform” includes the hardware configuration (e.g., Basic Input/Output System (BIOS) settings) and the hypervisor configuration. (This assumes that the hypervisor is running directly on the hardware, and not on top of another operating system. This also assumes that the hypervisor has not been compromised and that the hypervisor is the designated version.)
2. **Before each hypervisor launch, verify (measure) the trustworthiness of the cloud server platform.** The items configured in goal 1 (BIOS and hypervisor) need to have their configurations verified before launching the hypervisor to ensure that the assumed level of trust is still in place.
3. **During hypervisor execution, periodically audit the trustworthiness of the cloud server platform.** This periodic audit is essentially the same check as that performed as goal 2, except that it is performed frequently while the hypervisor is executing. Ideally this checking would be part of continuous monitoring.

Achieving all of these goals will not prevent attacks from succeeding, but will cause unauthorized changes to the hypervisor or BIOS to be detected much more rapidly than they otherwise would have been. So if a hypervisor is tampered with or subverted, the alteration will be detected quickly, almost instantly if continuous monitoring is being performed. This allows an immediate stop to execution, thus limiting damage to the information being processed within the cloud computing server.

For more information on the technical topics being addressed by these goals, see the following NIST publications:

- NIST Special Publication (SP) 800-125, *Guide to Security for Full Virtualization Technologies*
  http://csrc.nist.gov/publications/PubsSPs.html#800-125

- NIST SP 800-128, *Guide for Security-Focused Configuration Management of Information Systems*
  http://csrc.nist.gov/publications/PubsSPs.html#800-128

- NIST SP 800-137, *Information Security Continuous Monitoring for Federal Information Systems and Organizations*
  http://csrc.nist.gov/publications/PubsSPs.html#800-137

- NIST SP 800-144, *Guidelines on Security and Privacy in Public Cloud Computing*
  http://csrc.nist.gov/publications/PubsSPs.html#800-144

- NIST SP 800-147B, *BIOS Protection Guidelines for Servers*
  http://csrc.nist.gov/publications/PubsSPs.html#SP-800-147-B

- Draft NIST SP 800-155, *BIOS Integrity Measurement Guidelines*
  http://csrc.nist.gov/publications/PubsSPs.html#800-155

2.2.2 Stage 1: Trust-Based Homogeneous Secure Migration

Once stage 0 has been successfully completed, the next objective is to be able to migrate workloads among homogeneous, trusted platforms. Workload migration is a key attribute of cloud computing, improving scalability and reliability. The purpose of this stage is to ensure that any server that a workload is moved to will have the same level of security assurance as the server it was initially deployed to.

Stage 1 includes the following prerequisite goals:

1. **Deploy workloads only to cloud servers with trusted platforms.** This basically means that you perform stage 0, goal 3 (auditing platform trustworthiness during hypervisor execution) and only deploy a workload to the cloud server if the audit demonstrates that the platform is trustworthy.

2. **Migrate workloads on trusted platforms to homogeneous cloud servers on trusted platforms; prohibit migration of workloads between trusted and untrusted servers.** For the purposes of this publication, homogeneous cloud servers are those that have the same hardware architecture (e.g., Central Processing Unit (CPU) type) and the same hypervisor type, and that reside in the same cloud with a single management console. If a workload has been deployed to a trusted platform, the level of assurance can only be sustained if it is migrated only to hosts with comparable trust levels. So this goal is built upon stage 0, goal 3 (auditing platform trustworthiness during hypervisor execution) performed on both the workload's current server and the server to migrate the workload to. Only if both servers pass their audits can the migration be permitted to occur.

Achieving these goals ensures that the workloads are deployed to trusted platforms, thus reducing the chance of workload compromise.

For more information on the technical topics being addressed by these goals, see the following NIST publications:

- NIST SP 800-137, *Information Security Continuous Monitoring for Federal Information Systems and Organizations*
  http://csrc.nist.gov/publications/PubsSPs.html#800-137

- NIST SP 800-144, *Guidelines on Security and Privacy in Public Cloud Computing*
  http://csrc.nist.gov/publications/PubsSPs.html#800-144

- NIST SP 800-147B, *BIOS Protection Guidelines for Servers*
  http://csrc.nist.gov/publications/PubsSPs.html#SP-800-147-B

- Draft NIST SP 800-155, *BIOS Integrity Measurement Guidelines*
  http://csrc.nist.gov/publications/PubsSPs.html#800-155

2.2.3 Stage 2: Trust-Based and Geolocation-Based Homogeneous Secure Migration

The next stage builds upon stage 1 by adding the ability to continuously monitor and enforce geolocation restrictions.

Stage 2 includes the following prerequisite goals:

1. **Have trusted geolocation information for each trusted platform instance.** This information would be stored within the cloud server's cryptographic module (as a cryptographic hash within the hardware cryptographic module), so that it could be verified and audited readily.

2. **Provide configuration management and policy enforcement mechanisms for trusted platforms that include enforcement of geolocation restrictions.** This goal builds upon stage 1, goal 2 (migrating workloads on trusted platforms to other trusted platforms); it enhances stage 1, goal 2 by adding a geolocation check to the server to migrate the workload to.

3. **During hypervisor execution, periodically audit the geolocation of the cloud server platform against geolocation policy restrictions.** This goal is built upon stage 0, goal 3 (auditing platform trustworthiness during hypervisor execution), but it is specifically auditing the geolocation information against the policies for geolocation to ensure that the server's geolocation does not violate the policies.

Achieving these goals ensures that the workloads are not transferred to a server in an unsuitable geographic location. This avoids issues caused by clouds spanning different physical locations (e.g., countries or states with different data security and privacy laws).

For more information on the technical topics being addressed by these goals, see the following NIST publications:

- NIST SP 800-128, *Guide for Security-Focused Configuration Management of Information Systems*
  http://csrc.nist.gov/publications/PubsSPs.html#800-128

- NIST SP 800-137, *Information Security Continuous Monitoring for Federal Information Systems and Organizations*
  http://csrc.nist.gov/publications/PubsSPs.html#800-137

- NIST SP 800-147B, *BIOS Protection Guidelines for Servers*
  http://csrc.nist.gov/publications/PubsSPs.html#SP-800-147-B

- Draft NIST SP 800-155, *BIOS Integrity Measurement Guidelines*
  http://csrc.nist.gov/publications/PubsSPs.html#800-155

3 Usage Scenario Instantiation Example: Stage 0

This section describes stage 0 of the proof of concept implementation (platform attestation and safer hypervisor launch).

3.1 Solution Overview

This stage of the usage scenario enables the creation of what are called trusted compute pools. Also known as trusted pools, they are physical or logical groupings of computing hardware in a data center that are tagged with specific and varying security policies, and the access and execution of apps and workloads are monitored, controlled, audited, etc. In this phase of the solution, an attested launch of the platform including the hypervisor is deemed as a trusted node, and is added to the trusted pool.

Figure 1 depicts the concept of trusted pools. The resources tagged green indicate trusted ones. Critical policies can be defined such that security-sensitive cloud services can only be launched on these trusted resources, or migrated to other trusted platforms within these pools.

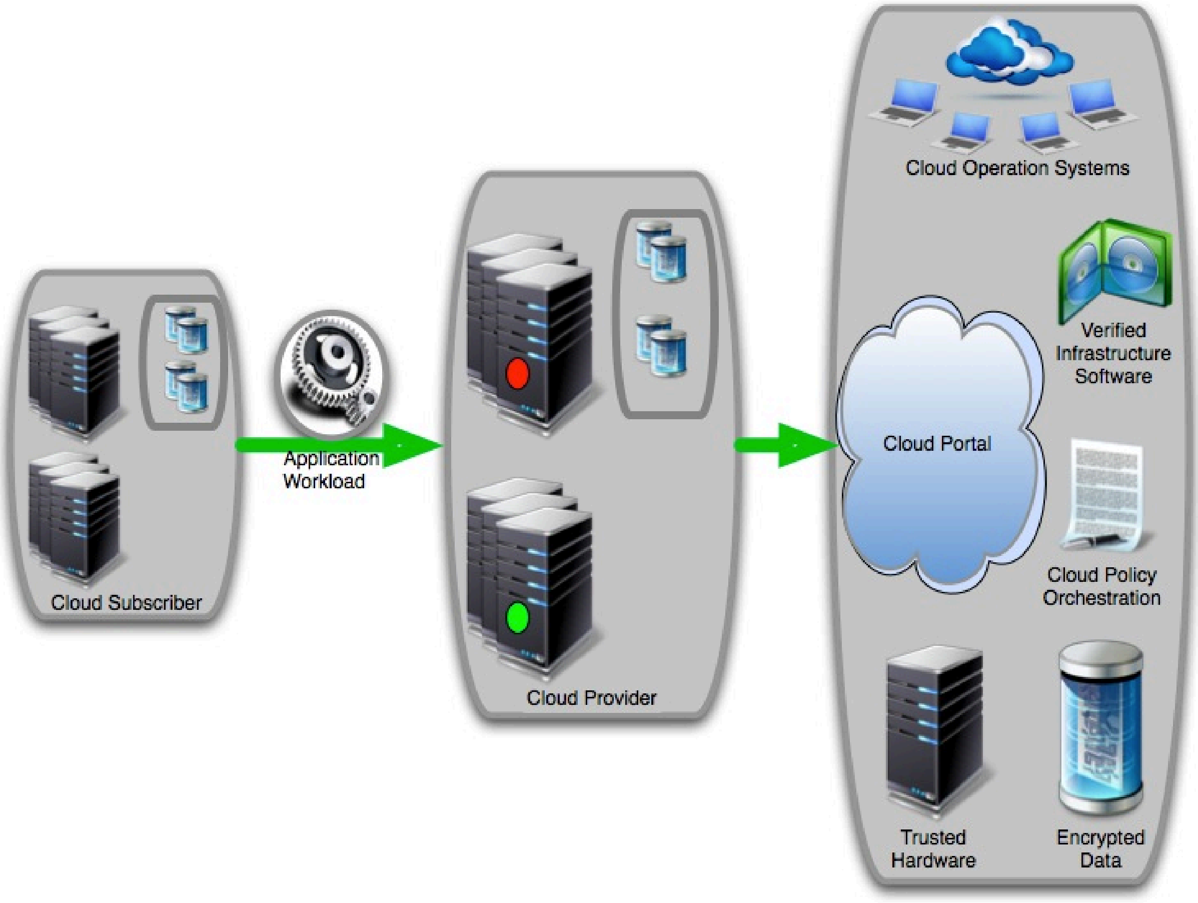


Figure 1: Concept of Trusted Pools

In order to have a trusted launch of the platform, the two key questions that should be answered are:

1. How would the entity needing this information know if a specific platform has the necessary enhanced hardware-based security features enabled and if a specific platform has a defined/compliant operating system (OS)/virtual machine manager (VMM) running on it?

2. Why should the entity requesting this information, which in a cloud environment would be a scheduler/orchestrator trying to schedule a workload on a set of available nodes/servers, believe the response from the platform?

Attestation provides the definitive answers to these questions. Attestation is the process of providing a digital signature of a set of measurements securely stored in hardware, then having the requestor validate the signature and the set of measurements. Attestation requires roots of trust. The platform has to have a Root of Trust for Measurement (RTM) that is implicitly trusted to provide an accurate measurement, and enhanced hardware-based security features provide the RTM. The platform also has to have a Root of Trust for Reporting (RTR) and a Root of Trust for Storage (RTS), and the same enhanced hardware-based security features provide these.

The entity that challenged the platform for this information now can make a determination about the trust of the launched platform by comparing the provided set of measurements with “known good/golden” measurements. Managing the “known good” for different hypervisors and operating systems, and various BIOS software, and ensuring they are protected from tampering and spoofing is a critical IT operations challenge. This capability can be internal to a service provider, or it could be delivered as a service by a trusted third party for service providers and enterprises to use.

3.2 Solution Architecture

Figure 2 provides a layered view of the solution system architecture. The indicated servers in the resource pool include a hardware module for storing sensitive keys and measurements. All the servers are configured by the virtualization management server.

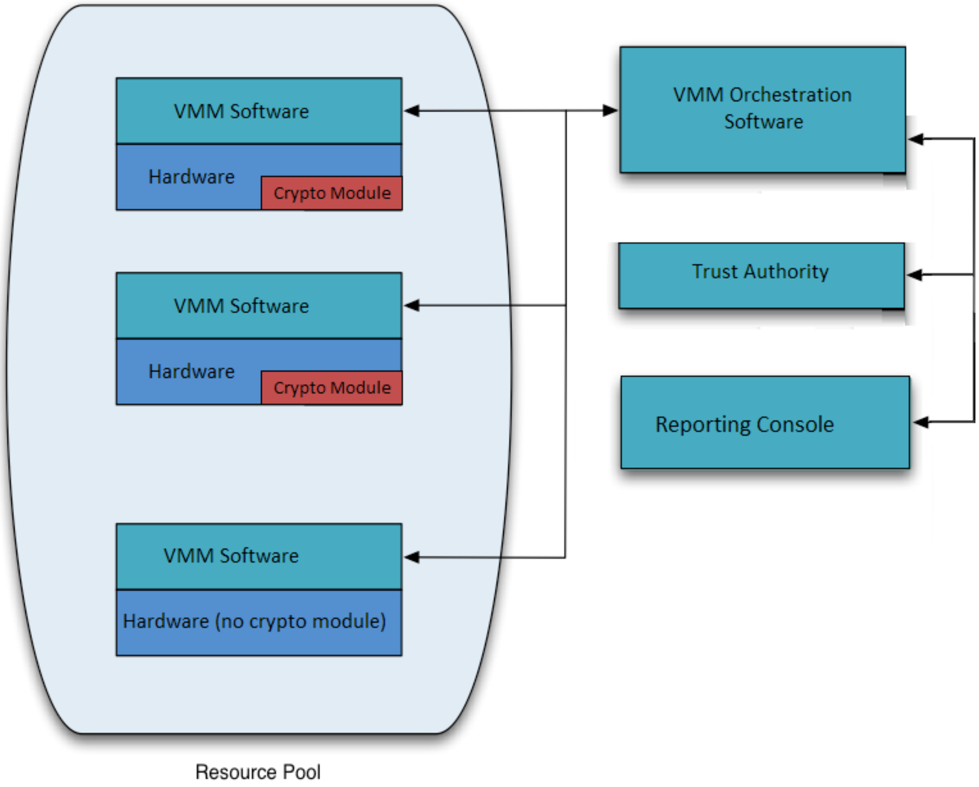


Figure 2: Stage 0 Solution System Architecture

The initial step in instantiating the architecture requires provisioning the server for enhanced hardware-based security features. This currently requires physical access to the server to access the BIOS, enable a set of configuration options to use the hardware module (including taking ownership of the module), and activate the enhanced hardware-based security features. This process is highly BIOS and OEM dependent. This step is mandatory for a measured launch of the OS/hypervisor.

Assuming that the virtual machine (VM) supports the enhanced hardware-based security features and these features have been enabled and a launch policy configured, the hypervisor undergoes a measured launch, and the BIOS and VMM components are measured (cryptographically) and placed into the server hardware module. These measurement values are accessible through the virtualization management server via the Application Program Interface (API). When the hosts are initially configured with the virtualization management server, the relevant measurement values are cached in the virtualization management database.

In addition to the measured launch, this solution architecture also provides provisions to assign a secure geolocation tag (geotag) to each of the servers during the provisioning process. The geotag is provisioned to a non-volatile index in the hardware module via an out-of-band mechanism, and on a hypervisor launch, the contents of the index are inserted/extended into the hardware module. Enhanced hardware-based security features provide the interface and attestation to the geotag information, including the geotag lookup and user-readable/presentable string/description.

4 Usage Scenario Instantiation Example: Stage 1

This section discusses stage 1 of the proof of concept implementation (trust-based homogeneous secure migration), which is based on the stage 0 work and adds components that migrate workloads among homogeneous, trusted platforms.

4.1 Solution Overview

Figure 3 shows the operation of the stage 1 solution. It assumes that Server A and Server B are two servers within the same cloud.

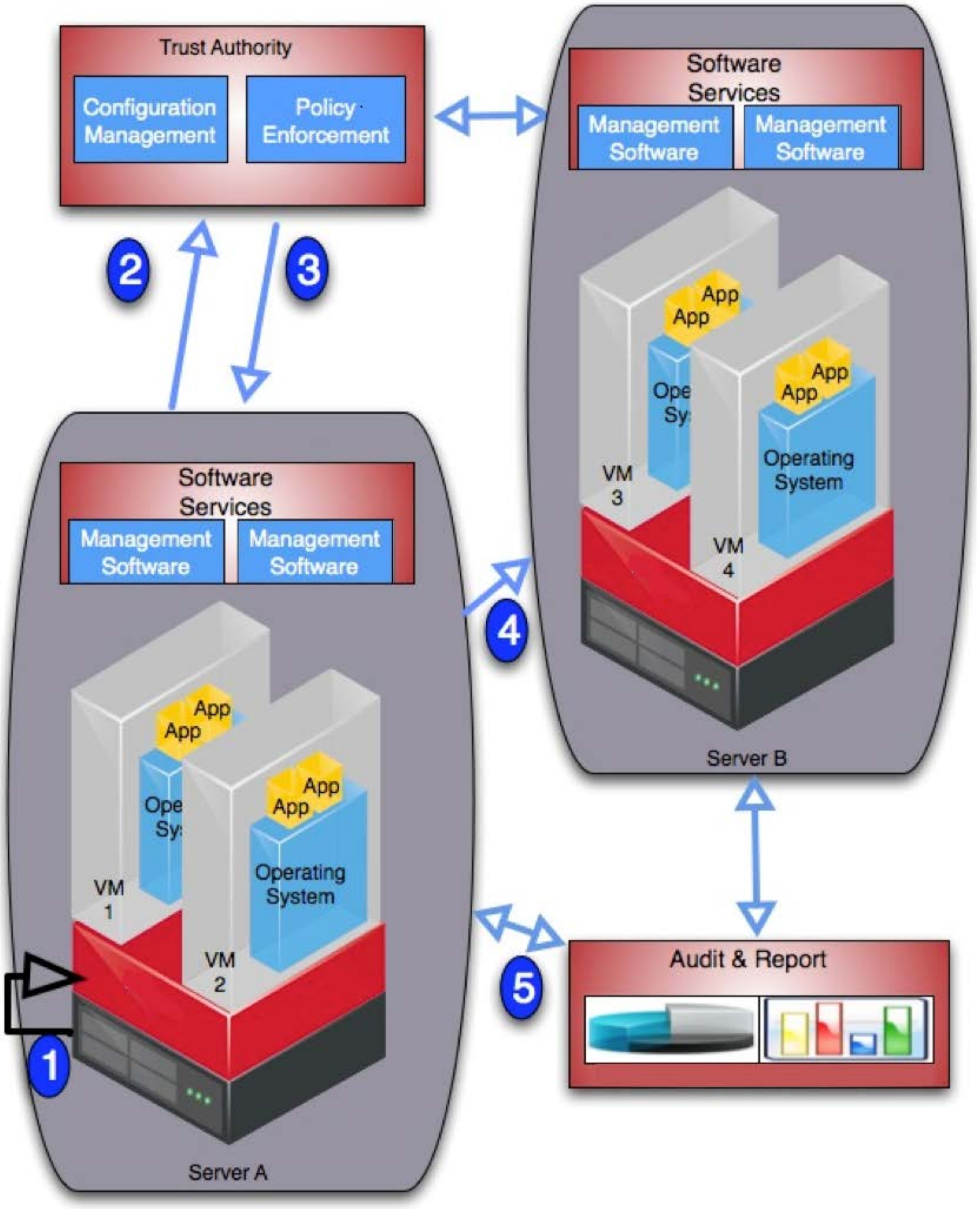


Figure 3: Stage 1 Solution Overview

There are five generic steps performed in the operation of the stage 1 solution, as outlined below and reflected by the numbers in Figure 3:

1. Server A performs a measured launch, with the enhanced hardware-based security features populating the measurements in the hardware module.

2. Server A sends a quote to the Trust Authority. The quote includes signed hashes of the BIOS, Trusted Boot (TBOOT), VM, and geotag values.

3. The Trust Authority verifies the signature and hash values and sends an authorization token to Server A.

4. Server A's management layer executes a policy-based action (in this case, a VM transfer to Server B).

5. Server A and Server B get audited periodically based on their measurement values.

4.2 Solution Architecture

The stage 1 architecture is identical to the stage 0 architecture (see Figure 2), with additional measurement occurring related to the migration of workloads among trusted hosts.

5 Usage Scenario Instantiation Example: Stage 2

This section discusses stage 2 of the proof of concept implementation (trust-based and geolocation-based homogeneous secure migration), which is based on the stage 1 work and adds components that take into account geolocation restrictions.

5.1 Solution Overview

Stage 2 adds the monitoring of measurements in a governance, risk, and compliance dashboard. One chart that might appear in such a dashboard could reflect the relative size of the pools of trusted and untrusted cloud servers. This could be displayed by percentage and/or count. Figure 4 shows a notional example.

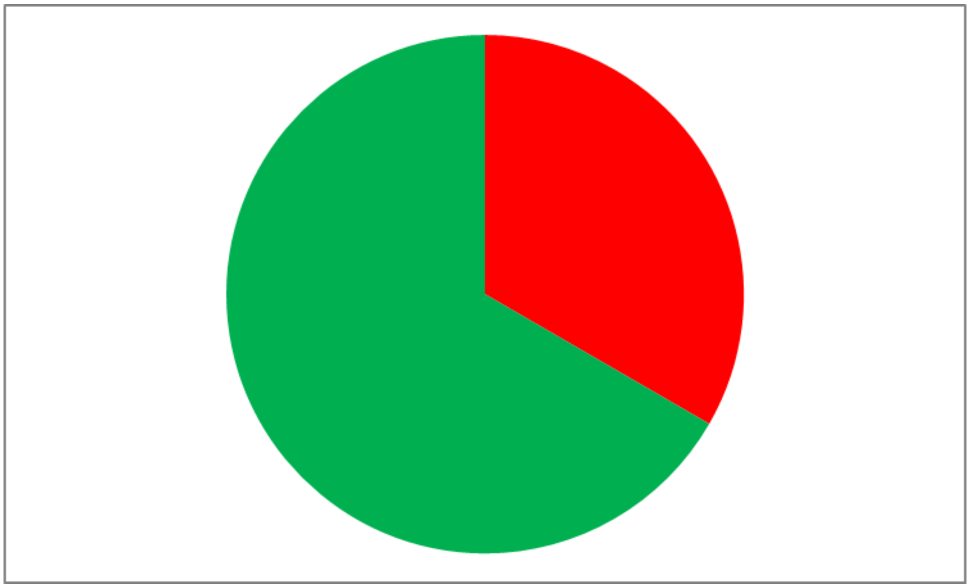


Figure 4: Notional Graph of Trusted and Untrusted Pool Sizes

Table 1 is a drill-down page from the high-level dashboard view shown in Figure 4. It provides more details on all the servers within the cloud. In this example, there are three servers. Information listed for each server includes the server's IP address, the status of the three measurements (trusted boot validation, geolocation validation, and system validation), and the timestamp for when those measurements were taken.

Table 1: Trusted Boot and Geolocation Compliance View

VM Host	Trusted Boot Validation	Geolocation Validation	System Validation	Last Data Pull
<Host 1>	Yes/No	Yes/No	Yes/No	<Timestamp>
<Host 2>	Yes/No	Yes/No	Yes/No	<Timestamp>
<Host 3>	Yes/No	Yes/No	Yes/No	<Timestamp>

Figure 5 shows a drill-down from Table 1 for an individual server. It includes the raw measurement data for the trusted boot validation and the geolocation validation, alongside the "golden values" that the trusted boot value and geolocation value are expected to have. It also shows when the server was first measured and when it was most recently measured. Measuring each server's characteristics frequently (such as every five minutes) helps to achieve a continuous monitoring solution for the servers.

General Information			
VM Host:	<IP Address or Host Name>	Tracking ID:	<Unique ID>
First Published:	<Time Stamp>	Last Data Pull:	<Time Stamp>
Golden Trusted Boot Value:	<Provision Time Trusted Boot Value>	Current Trusted Boot Value:	<Current Trusted Boot Value>
Golden Geolocation Value:	<Provision Time Geolocation Value>	Current Geolocation Value:	<Current Geolocation Value>

Trust Validation			
System Validation:	Yes/No		
Trusted Boot Validation:	Yes/No	Geolocation Validation:	Yes/No

Figure 5: Single Server Overview

5.2 Solution Architecture

The stage 2 architecture is identical to the stage 0 and stage 1 architectures (see Figure 2), with additional reporting and monitoring occurring related to geolocation.

Appendix A—Hardware Architecture and Prerequisites

This appendix provides an overview of the high-level hardware architecture of the proof of concept implementation, as well as details on how Intel platforms implement hardware modules and enhanced hardware-based security functions.

A.1 High-Level Implementation Architecture

Following the recommendations proposed in NIST SP 800-125, the high-level architecture of the proof of concept implementation is composed of three distinct networks to isolate the traffic flowing through the management VMs, storage device, and public VMs. Figure 6 represents the proof of concept implementation architecture, which includes the various hardware and logical networks.

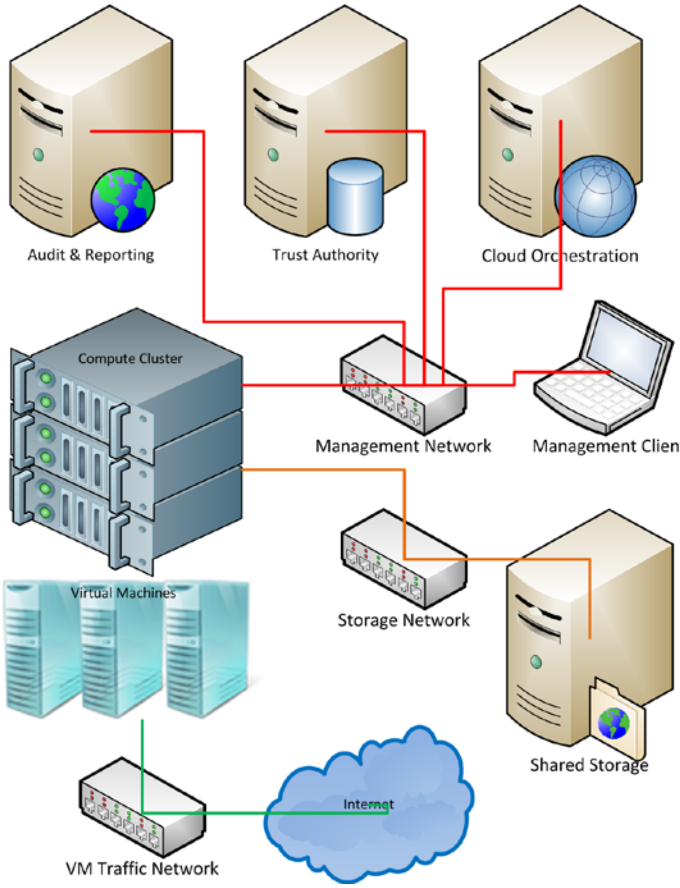


Figure 6: Proof of Concept Implementation

Management Network

The management workstation is connected to the management network, which includes the four management servers. A dedicated server is used to host the management VMs for the other management servers: the cloud orchestration server, the trust authority server, and the audit and reporting server. The cloud orchestration server manages the remaining three servers, which are part of the cluster hosting the public VMs. The measurement server takes measurements of the trusted cloud cluster and directs them to the cloud orchestration server. The audit and reporting server communicates with the cloud orchestration server to obtain the measurement values to reflect in the dashboard view.

The management network is connected to a dedicated non-routable network. An additional non-routable network is used to support the automated migration of the VMs from different nodes across the trusted cluster.

Storage Network

The storage device provides shared storage where the public VMs are hosted. The three public VM servers are connected to the storage network, which uses a non-routable network.

Public VM Network

The public VM network is accessible to the workload owners from the Internet. In the demonstration, a single server represents a typical public workload VM controlled by the customers over the Internet. A dedicated network card on each of the trusted cluster server nodes is used to carry the VM's traffic.

A.2 Intel Trusted Execution Technology (Intel TXT) & Trusted Platform Module (TPM)

Hardware-based root of trust, when coupled with an enabled operating system, hypervisor and solutions, constitutes the foundation for a more secure computing platform. This secure platform ensures OS and VMM integrity at boot from rootkits or other low-level attacks. It establishes the trustworthiness of the server and 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 foundational elements of a single platform. These 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. This is the Core-RTM (CRTM). An RTM is the first component to send integrity-relevant information (measurements) to the RTS. Trust in this component is the basis for trust in all the other measurements. 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.

Intel Trusted Execution Technology (Intel TXT) is the RTM, and it is a mechanism to enable visibility, trust, and control in the cloud. 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 the face of increasingly hostile environments.

Intel TXT incorporates a number of secure processing innovations, 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 only be decrypted by the same environment that encrypted it.

- **Attestation.** Enables a system to provide assurance that the protected environment has been correctly invoked and to take 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.

Intel Xeon processor 5600 series and the more recent Xeon Processor E3, Xeon Processor E7, and forthcoming Xeon Processor E5 series processors support Intel TXT.

Figure 7 depicts the different hardware and software components that Intel TXT is comprised of. 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 any code that does not match or, alternately, indicate 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.

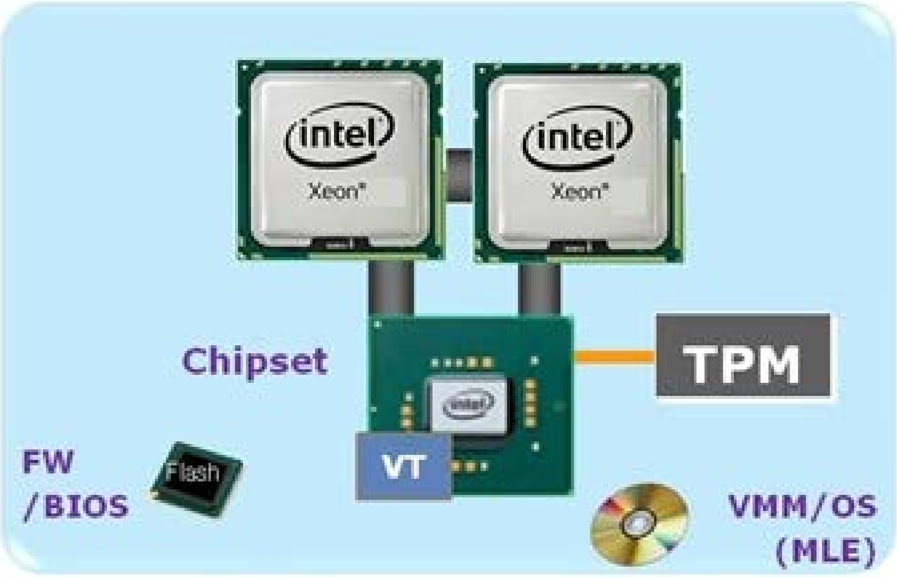


Figure 7: Intel TXT Components

Figure 8 illustrates two different scenarios. In the first, the measurements match the expected values, so the launch of the BIOS, firmware, and VMM is allowed. In the second, the system has been compromised by a rootkit (malicious hypervisor), which attempts to install itself below the hypervisor to gain access to the platform. In this case, the Intel TXT-enabled, MLE-calculated hash system measurement will differ from the expected value due to the insertion of the rootkit. Therefore, based on the launch policy, Intel TXT could abort the launch of the hypervisor or report an untrusted launch to the virtualization or cloud management infrastructure for subsequent use.

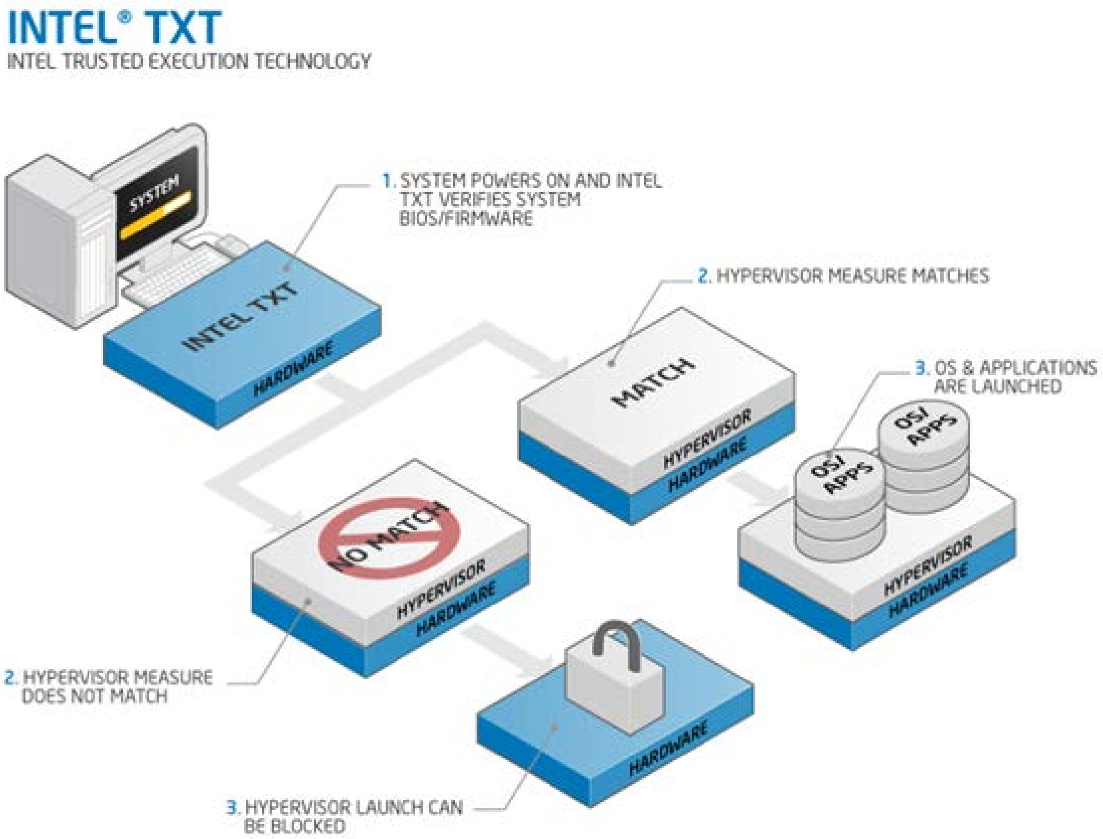

INTEL® TXT
INTEL TRUSTED EXECUTION TECHNOLOGY
SYSTEM
1. SYSTEM POWERS ON AND INTEL TXT VERIFIES SYSTEM BIOS/FIRMWARE
INTEL TXT
HARDWARE
2. HYPERVISOR MEASURE MATCHES
MATCH
HYPERVISOR
HARDWARE
3. OS & APPLICATIONS ARE LAUNCHED
OS/ APPS
OS/ APPS
HYPERVISOR
HARDWARE
NO MATCH
HYPERVISOR
HARDWARE
2. HYPERVISOR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

Figure 8: How Intel TXT Protects the Launch Environment

A.3 Attestation

There are two main considerations for usage scenarios to be instantiated and delivered in a cloud:

- How would the entity needing this information know if a specific platform has Intel TXT enabled or if a specific server has a defined or compliant BIOS or OS 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 authority provides the definitive answers to these questions. Attestation up-levels the notion of roots of trust by making the information from various roots of trust visible and usable by other entities. Figure 9 illustrates the attestation protocol providing the means for conveying measurements to the challenger. The endpoint attesting device must have a means of measuring the BIOS firmware, low level device drivers, and operating system and virtual machine monitor components, and forwarding those measurements to the attestation authority. The attesting device must do this while protecting the integrity, authenticity, nonrepudiation, and in some cases, confidentiality of those measurements.

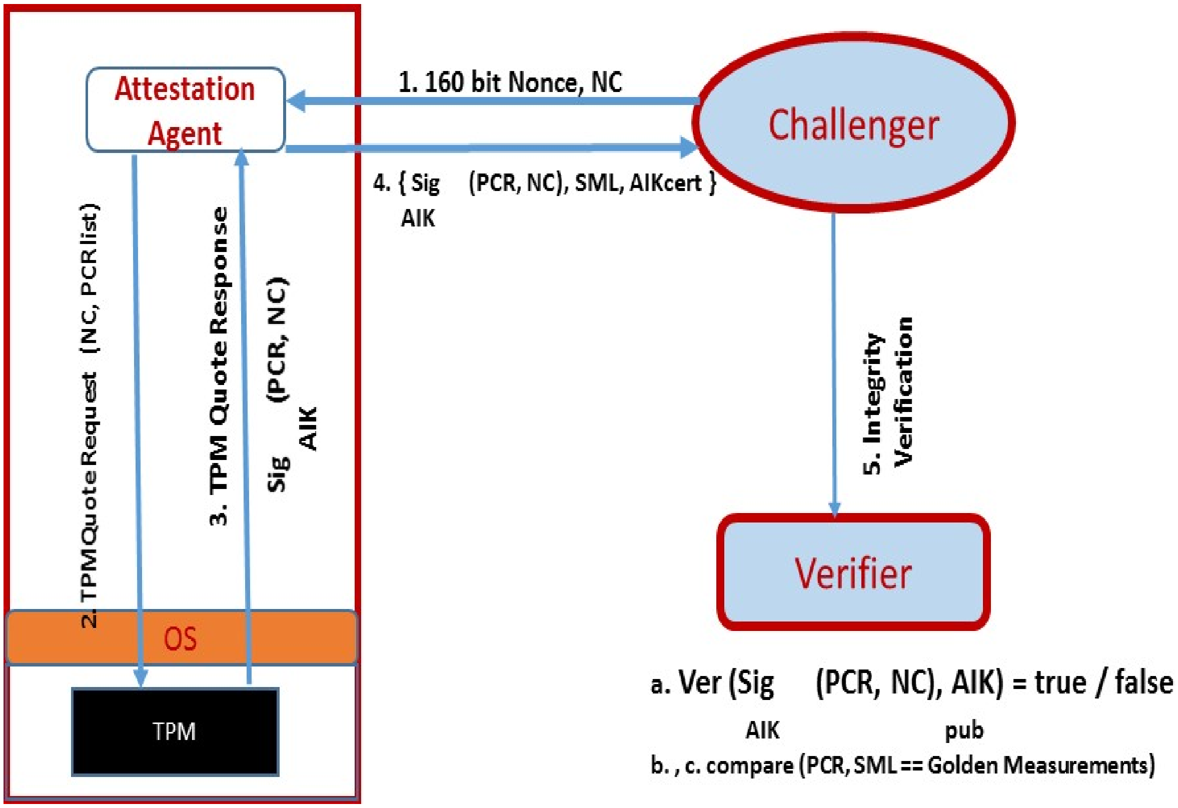

Attestation Agent
1. 160 bit Nonce, NC
Challenger
4. { Sig AIK (PCR, NC), SML, AIKcert }
2. TPMQuoteRequest (NC, PCR list)
3. TPM Quote Response
Sig AIK (PCR, NC)
5. Integrity Verification
Verifier
OS
TPM
a. Ver (Sig AIK (PCR, NC), AIK pub) = true / false
b. , c. compare (PCR, SML == Golden Measurements)

Figure 9: Remote Attestation Protocol

Here are the steps shown in Figure 9 for the remote attestation protocol:

- In step 1, the challenger, at the request of a requester, creates a non-predictable nonce (NC) and sends it to the attestation agent on the attesting node, along with the selected list of Platform Configuration Registers (PCRs).
- In step 2, the attestation agent sends that request to the TPM as a TPMQuoteRequest with the nonce and the PCR List.
- In step 3, in response to the TPMQuote request, the TPM loads the attestation identity key from protected storage in the TPM by using the storage root key (SRK), and performs a *TPM Quote* command, which is used to sign the selected PCRs and the provided nonce (NC) with the private key *AIKpriv*. Additionally, the attesting agent retrieves the stored measurement log (SML).
- In step 4, the *integrity response* step, the attesting agent sends the response consisting of the signed quote, signed nonce (NC), and the SML to the challenger. The attesting agent also delivers the Attestation Identity Key (AIK) credential, which consists of the AIKpub that was signed by a privacy Certificate Authority (CA).
- In step 5a, the challenger validates if the AIK credential was signed by a trusted privacy CA, thus belonging to a genuine TPM. The challenger also verifies whether AIKpub is still valid by checking the certificate revocation list of the trusted issuing party.
- In step 5b, the challenger verifies the signature of the quote and checks the freshness of the quote.

- In step 5c, based on the received SML and the PCR values, the challenger processes the SML, compares the individual module hashes that are extended to the PCRs against the "good known or golden values," and recomputes the received PCR values. If the individual values match the golden values and if the computed values match the signed aggregate, the remote node is asserted to be in a trusted state.

This protocol is highly resistant against replay attacks, tampering, and masquerading.

Appendix B—Platform Implementation: HyTrust

This section contains supplementary information provided by HyTrust describing all the required components and steps required to set up the proof of concept implementation.

B.1 Solution Architecture

Figure 10 shows the architecture depicted in Appendix A, but with the specific products used in the HyTrust platform implementation.

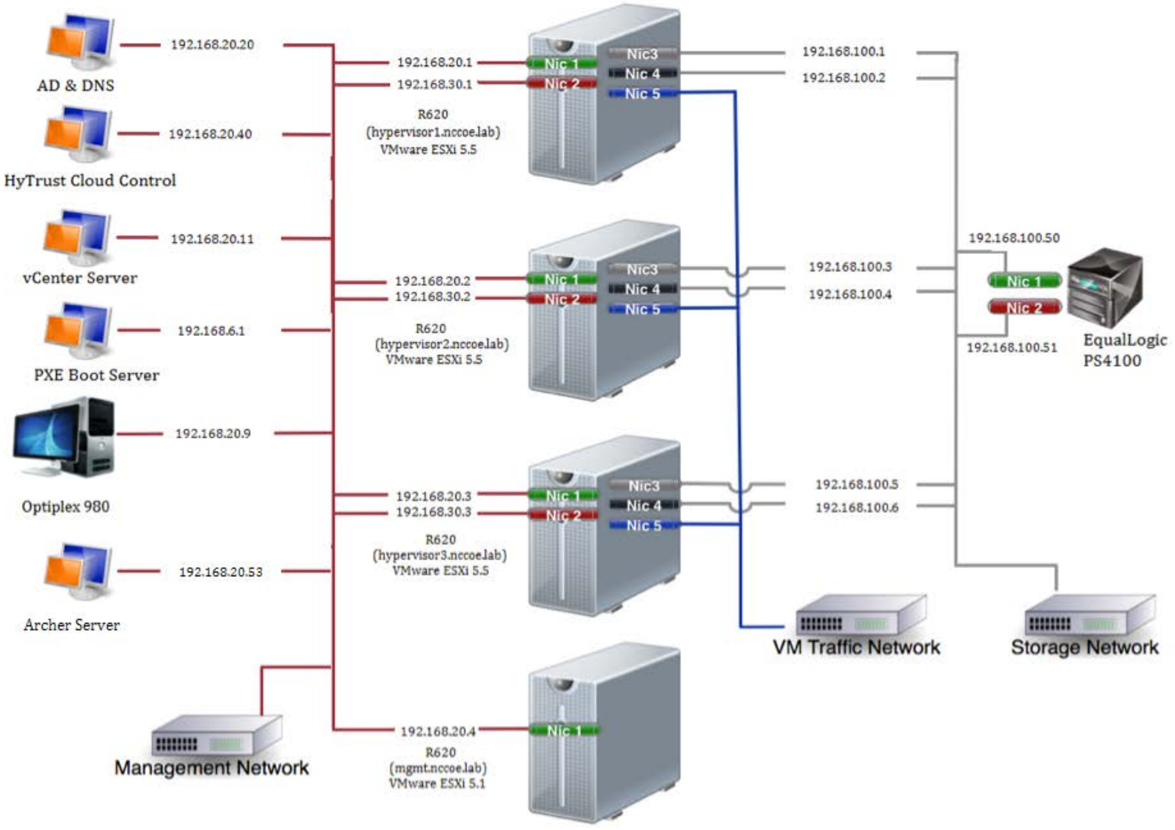


Figure 10: HyTrust Proof of Concept Implementation

B.2 Hardware Description

The implemented architecture is composed of three Dell servers running VMware ESXi 5.5 configured as a cluster with a shared resource pool utilizing an iSCSI storage device, a management node that includes five VMs providing different functionalities, and a dedicated management workstation.

Trusted Cloud Cluster:

- 3 x Dell PowerEdge R620 (Intel TXT enabled):
  - 2 x Intel Xeon Processor E5-2660 @ 2.20 GHz
  - 64 GB Memory
  - VMware ESXi 5.5 hosting the following VMs:
    - Windows Server 2008 R2 for test workload VM connected to the VM Traffic Network

Storage:

- Dell EqualLogic PS4100

Management Node:

- Dell PowerEdge R620 (Intel TXT enabled):
  - 2 x Intel Xeon Processor E5-2660 @ 2.20 GHz
  - 64 GB Memory
- VMware ESXi 5.1 hosting the following VMs:
  - Windows Server 2008 R2 with VMware vCenter Enterprise Plus Server
  - Windows Server 2008 R2 with Active Directory and DNS enabled
  - Ubuntu 12.04 set up as a PXE Boot Server (iPXE, tfptd, nfs)
  - HyTrust CloudControl
  - Windows Server 2008 R2 with RSA Archer

Management Workstation:

- Dell Optiplex 980
  - Windows 7 with VMware vSphere client

B.3 BIOS Changes

The following changes are required in the BIOS settings:

1. Set Intel TXT to “Enabled”.
2. Set Intel Virtualization Technology (Intel VT) to “Enabled”.
3. Set Intel VT for Directed I/O to “Enabled”.
4. Set “Administrator Password” and reboot prior to enabling the TPM.
5. Change TPM Administrative Control to “Turn On”; TPM state will show as “enabled and activated” after reboot.

B.4 HyTrust CloudControl Installation and Configuration with VMware Components

HTCC 4.1.0 Product Documentation:

http://downloads.hytrust.com/product_documentation/4.1.0/HyTrust_CloudControl_Installation_Guide.pdf

http://downloads.hytrust.com/product_documentation/4.1.0/HyTrust_CloudControl_Administration_Guide.pdf

HTCC 4.1.0 Prerequisites:

Technical Requirements for HyTrust CloudControl Appliance (HTCC)

Table 2: HTCC System Requirements

Minimum Requirement	HTCC Virtual Machine
Disk Space	30 GB*
Memory	4 GB**
Virtual CPU	4
Network	1 Network Interface Controller (NIC) minimum

* Thin provisioning for test environments only
** By default HTCC is deployed with 16 GB of RAM; for small test environments ONLY this can be changed to 4 GB of RAM prior to first power on.

IP address requirements <u>on the VM Management Network</u> (to be configured on the HTCC):

- The HTCC itself needs one IP address (Eth 0 Interface)
- One IP address for the vCenter Server(s) that will be protected by the HTCC
- One IP address for the vCenter Web Client Server (if applicable) that will be protected by the HTCC
- One IP address for each ESX or ESXi host that will be protected by the HTCC. For example, if you have 10 hosts to protect, a total of 12 IP addresses will be required: 1 HTCC + 1 vCenter + 10 hosts. HTCC Management IP and PIPs (Published IP addresses) have to be on the same subnet.

Authentication:

- Root credentials for all ESX or ESXi hosts
- Administrator-level account for the vCenter Server instance (Service Account is recommended) typically named "htccVCserviceaccount"
- Domain user account to the Active Directory (AD) environment used for testing (a dedicated AD account for HTCC is recommended) typically named "htccADserviceaccount"

Active Directory Groups:

- ASC_SuperAdmins
- ASC_NetworkAdmins
- ASC_DCAdmins

Active Directory Users:*

- SuperAdminUser
- NetworkAdminUser
- DCAdminUser

* Be sure to add the users to the corresponding groups.

VMware Components:

- ESXi = 5.5 Update 1 build 1623387
- vCenter = 5.5.0 Update 1 build 1623101

B.5 Trust Authority Installation and Configuration

Figure 11 explains the necessary steps in order to provision HTCC PolicyTags for geolocation. Each step has a detailed writeup in this subsection or the following subsection of the appendix.

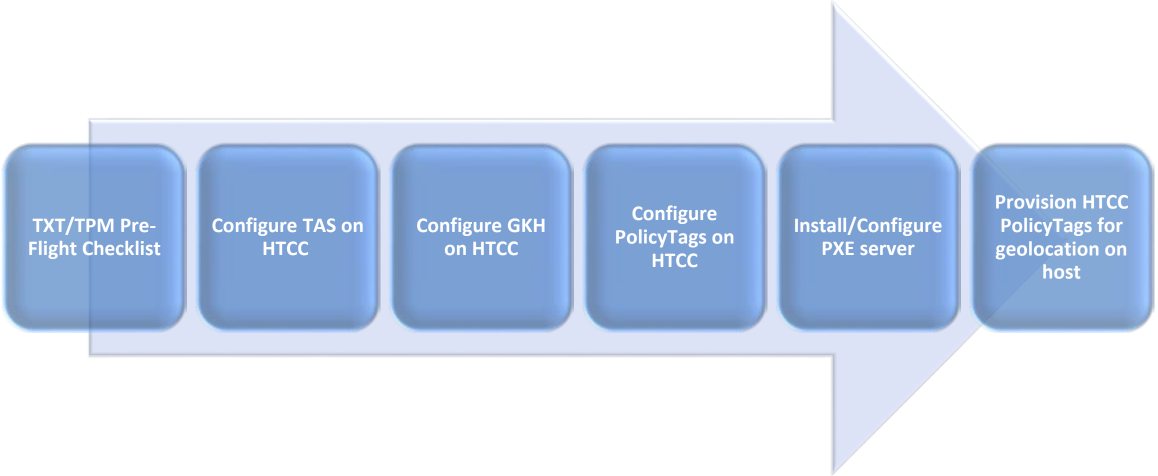

TXT/TPM Pre-Flight Checklist
Configure TAS on HTCC
Configure GKH on HTCC
Configure PolicyTags on HTCC
Install/Configure PXE server
Provision HTCC PolicyTags for geolocation on host

Figure 11: Process for Provisioning HTCC PolicyTags for Geolocation

B.5.1 TXT/TPM Pre-Flight Checklist

- Verify TXT/TPM are enabled properly in the BIOS of the hosts.
- Verify hypervisor has taken ownership of the TPM on all hosts from local host command line; enter this command for ESXi, **esxcli hardware trustedboot get**. (Note: If either the Dynamic Root of Trust Measurement (DRTM) or TPM shows as false, please verify that TXT and TPM are enabled properly.)
- Verify all hostnames are lower case.
- Verify hosts domain is lower case, and add if blank.
- Verify DNS entries Forward and Reverse lookup zones are correct and with lower case. (Note: If DNS A records were repopulating in Microsoft DNS with UPPERCASE, this has not caused any issues.)
- Verify time on vCenter, ESXi hosts, and HTCC are in sync and within five minutes of each other.
- Verify VMM and BIOS versions in vCenter and PCR values in the vCenter Managed Object Browser (MOB). To navigate to these values, a series of links must be clicked in the MOB:

  1. content – content
  2. rootFolder – group-<ID>
  3. childEntity – datacenter-<ID>

4. hostFolder – group-<ID>
5. childEntity – domain-<ID>
6. host – host-<ID>

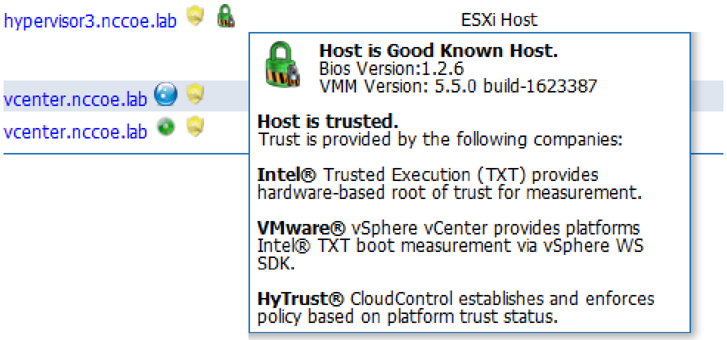
- To view the PCR Values, you can click on "runtime" or the Method, "**QueryTpmAttestationReport**" and click Invoke Method.
- If the Host is setup correctly and **supports TPM** and the vCenter Server has the appropriate PCR values a **long** page with many details will be launched.
- If the Host **does not support TPM** or the vCenter Server has no PCR data, only a few return types will be returned but no corresponding values.

B.5.2 Configure TAS on HTCC

To configure the Trust Attestation Service (TAS) on HTCC, please refer to the HyTrust CloudControl Administration Guide in the section titled "Configuring the Trust Attestation Service".

B.5.3 Configure GKH on HTCC

To configure Good Known Host (GKH) on HTCC, please refer to the HyTrust CloudControl Administration Guide in the section titled "Enabling Good Known Host". Figure 12 illustrates how the HTCC host dashboard displays GKH with the green lock icon. More details, such as BIOS version and VMM version, are available when the user hovers the mouse over the lock icon.



Host is Good Known Host.
Bios Version:1.2.6
VMM Version: 5.5.0 build-1623387

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 platforms Intel® TXT boot measurement via vSphere WS SDK.

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

Figure 12: VMware Host Selected as GHK

B.6 Trust Authority Registration

B.6.1 Configure PolicyTags on HTCC

To configure PolicyTags on HTCC, please refer to the HyTrust CloudControl Administration Guide in the section titled "PolicyTags".

B.6.2 Install/Configure PXE Server

PXE stands for Pre-boot eXecution Environment. PXE allows you to boot systems without the presence of a traditional operating system. In order to use PXE, first set up a boot-server and configure it for DHCP, TFTP, and NFS services. The following steps describe the boot-server setup process:

1. Set up a virtual machine as a boot server.
2. Set up the base operating system.
3. Set up services.
4. Configure GPXE to boot up on iPXE.

Prerequisites:

- VMware ESXi 5.0 or later
- New virtual machine
- vHW8
- Linux: Ubuntu Linux (64 bit) x86_64 or CentOS
- 1 vCPU
- 512 MB RAM
- 32 GB HD + LSI Logic Serial Attached SCSI (SAS) Host Bus Adapter (HBA)

Note: Disk can be as small as 4 GB, if only NFS mounting a remote filesystem.

Network Requirements:

Connecting the system to the bootstrap network can be accomplished in one of three ways:

- Set the physical switch ports on the upstream switch to access mode with manually relocated/ reconnected cabling. This can be used for the environment with a small number of machines.
- The upstream switch port that is connected to the physical uplink is configured as a trunk port. The virtual switch itself is inspecting and adding or removing the VLAN tags. The bootstrap services operate on the untagged native VLAN and all other VLANs are delivered tagged.
    - VMware reference Virtual Switch Tagging (VST) on how the network is set up in the lab: http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1003806#estPoints
- Use a "DHCP-Relay" or "DHCP-Helper" in combination with the Virtual Local Area Network (VLAN) trunk, with the actual bootstrap services operating on some other remote VLAN.

The PXE "Services VM" needs to have in-guest 'eth1' (Network Adapter 2 at VM configuration level) connected to a vSwitch or DvSwitch portgroup mapped to the same bootstrap network VLAN as described above. It does not need Promisc or Media Access Control (MAC) Spoof vSwitch permissions in order to function properly.

Set Up Services:

You will receive a '.tgz' bundle from HyTrust DevOps, along with pointers on where to obtain the correct version of the Intel Cloud Integrity Technology Asset Tag Provisioning ISO image.

Copy the bundle and ISO image into the home directory of the maintenance user (via SCP or SFTP), then extract the bundle:

1. tar -xzvf./path/to/file.tgz
2. Next, launch the configuration script within the extracted files:
3. SVC_VM_ALL=1./Services_VM/Services_VM_Configuration.sh
4. The script will install all requisite services (...) and move configuration files into place as shown in the next section.

Configure PXE Server for Local Network Topology:

Within the PXE "Services VM", the configuration files interoperate as follows:

- /etc/network/interfaces — eth0 / eth1 network interface configurations.
- /etc/default/isc-dhcp-server — DHCP daemon configuration.
- /etc/dhcp/dhcpd.conf — DHCP daemon configuration. Much of the file is comments.
- /etc/default/tftpd-hpa — TFTP daemon configuration.
- /etc/default/nfs-kernel-server — NFS daemon configuration.
- /etc/exports.d/local-intel.exports — NFS daemon filesystem export declaration.
- /var/lib/tftpboot/pxelinux.cfg/default — First phase (gPXE) bootstrap configuration.
- /var/lib/tftpboot/Intel/Mt.Wilson_TAS/2.0_GA/ATM/iPXE.cfg — Second phase (iPXE) bootstrap configuration.

Configure GPXE to Boot Up on iPXE:

Configure the iPXE server for Asset Tag management.

You will have to provide variables such as:

atag-server = "http://<HTCC management IP address>: <7443>/mtwilson/v2"

atag-username = 'tagadmin'

Provide four additional variables to specify where the casper boot loader will be located:

- nfs-host
- nfs-root
- http-host
- http-root

Provision HTCC PolicyTags for Geolocation on Host:

To provision HTCC PolicyTags for geolocation on host, please refer to the HyTrust CloudControl Administration Guide in the section titled "Provisioning Hosts".

Figure 13 depicts the PolicyTags Workflow from the HyTrust CloudControl Administration Guide in the section titled "Provisioning Hosts".

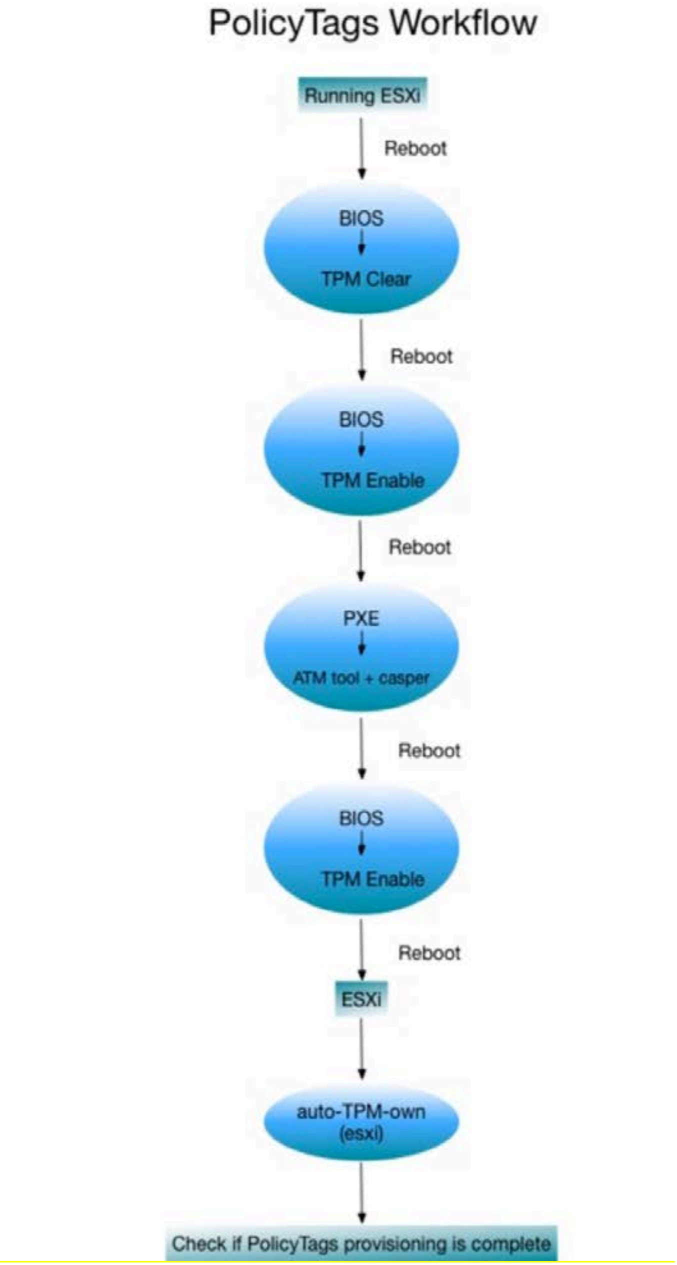
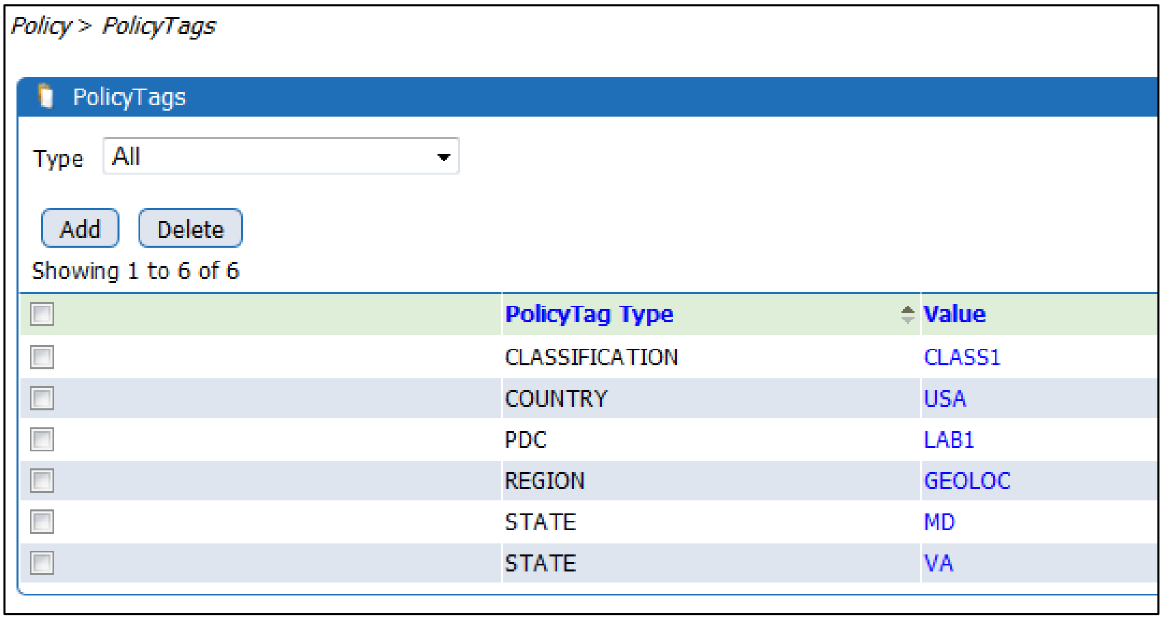


Figure 13: PolicyTags Provisioning Workflow in the HyTrust Environment

Figure 14 illustrates how to create different values for policy tags inside of the HTCC. These values are what make up the policy tags that get provisioned to individual hosts during the provisioning process.



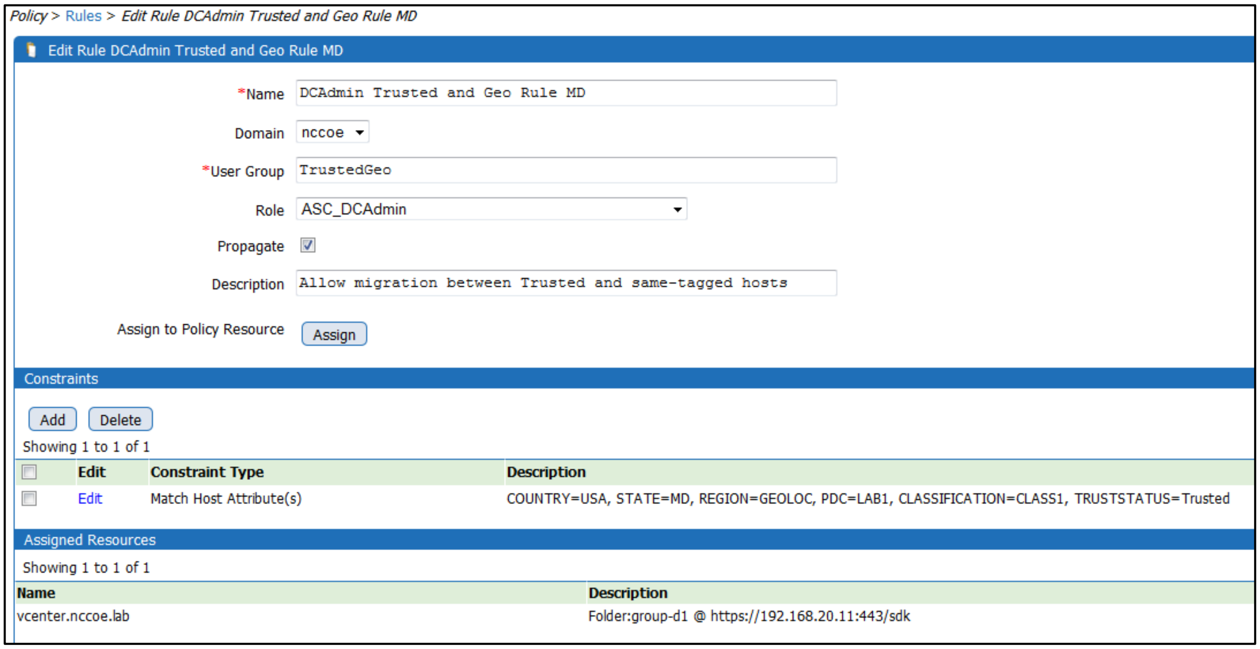
Type: All

Showing 1 to 6 of 6

□	PolicyTag Type	Value
□	CLASSIFICATION	CLASS1
□	COUNTRY	USA
□	PDC	LAB1
□	REGION	GEOLOC
□	STATE	MD
□	STATE	VA

Figure 14: PolicyTags Provisioning Workflow in the HyTrust Environment

Once policy tags are created, rules in the HTCC must be created so that the desired rules for the policy tags are enforced. Figure 15 shows rule creation that will allow for virtual machine migration between hosts that have a policy tag where the country is USA, State is MD, region is GEOLOC, classification is CLASS1, and trust status is Trusted. This rule will allow virtual machine migration between hosts that are trusted and within the same geolocation.


Policy > Rules > Edit Rule DCAdmin Trusted and Geo Rule MD
Edit Rule DCAdmin Trusted and Geo Rule MD
Assign
Constraints
Add
Delete
Assigned Resources

*Name: DCAdmin Trusted and Geo Rule MD

Domain: nccoe

*User Group: TrustedGeo

Role: ASC_DCAdmin

Propagate: ☑

Description: Allow migration between Trusted and same-tagged hosts

Assign to Policy Resource

Showing 1 to 1 of 1

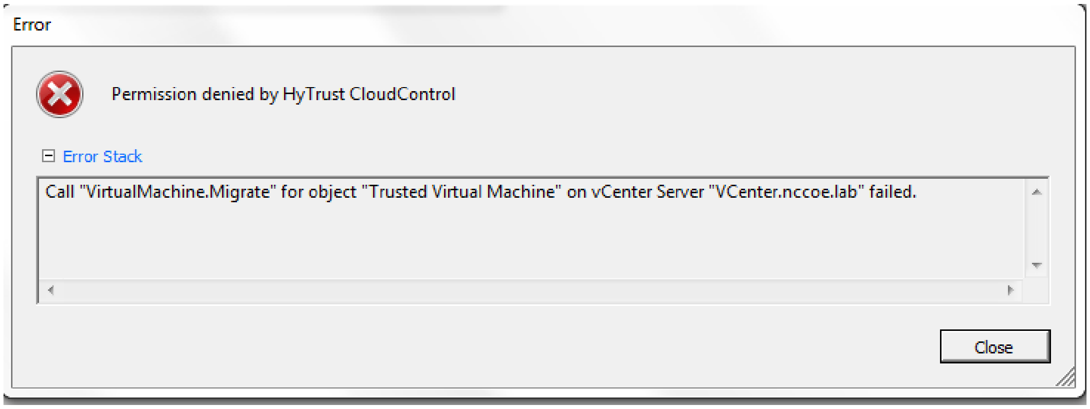
□	Edit	Constraint Type	Description
□	Edit	Match Host Attribute(s)	COUNTRY=USA, STATE=MD, REGION=GEOLOC, PDC=LAB1, CLASSIFICATION=CLASS1, TRUSTSTATUS=Trusted

Showing 1 to 1 of 1

Name	Description
vcenter.nccoe.lab	Folder:group-d1 @ https://192.168.20.11:443/sdk

Figure 15: Rule Creation to Enforce Policy Tags in HTCC

Once the rules have been created and applied in the HTCC, enforcement of these rules will automatically happen when a user logs into the HyTrust protected vCenter Server. Figure 16 shows the error message vCenter will display when a user tries to begin a virtual machine migration that does not meet the policy rules that are in place.



Permission denied by HyTrust CloudControl

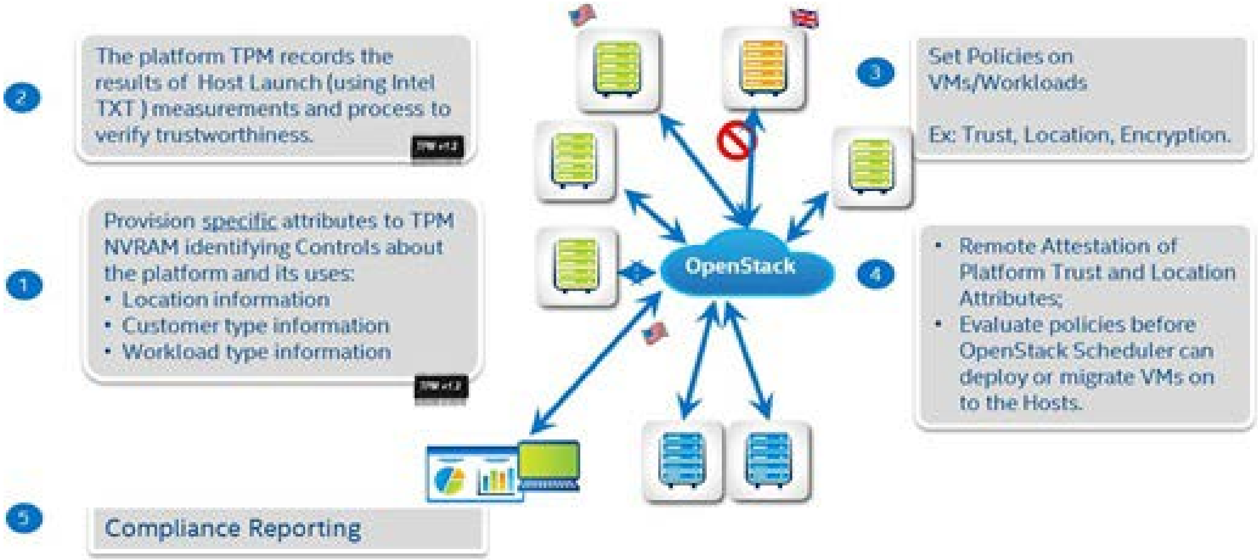
Call "VirtualMachine.Migrate" for object "Trusted Virtual Machine" on vCenter Server "VCenter.nccoe.lab" failed.

Figure 16: HTCC Policy Enforcement within vCenter

Appendix C—Platform Implementation: OpenStack

This section contains supplementary information provided by Intel describing all the required components and steps required to setup the proof of concept implementation.

Figure 17 details how geo and asset tagging can be incorporated and taken advantage of in OpenStack clouds to provide location and boundary control of workloads/OpenStack images. With geotags/asset tags, you can enforce policies to control placement, migration, or bursting of workloads to trusted systems in specific geographical locations or boundaries, and provide visibility and compliance of your workload policies to ensure tenants of compliance to trust and location policies.



2 The platform TPM records the results of Host Launch (using Intel TXT) measurements and process to verify trustworthiness.

1 Provision specific attributes to TPM NVRAM identifying Controls about the platform and its uses:
- Location information
- Customer type information
- Workload type information

3 Set Policies on VMs/Workloads

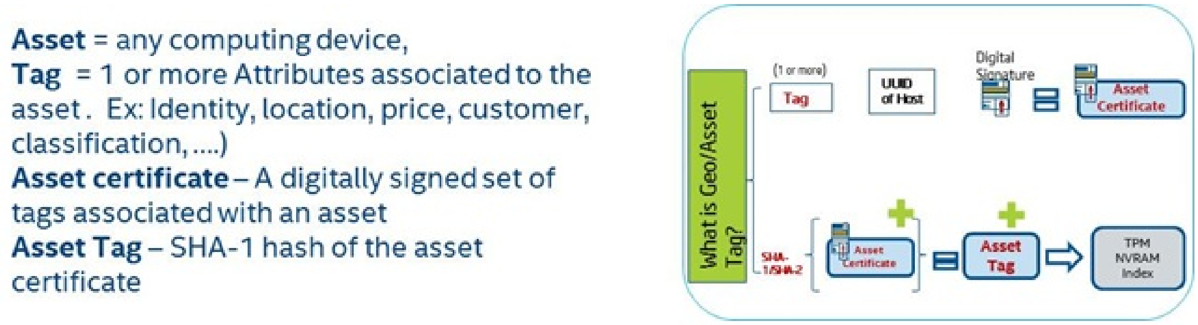
Ex: Trust, Location, Encryption.

4
- Remote Attestation of Platform Trust and Location Attributes;
- Evaluate policies before OpenStack Scheduler can deploy or migrate VMs on to the Hosts.

5 Compliance Reporting

Figure 17: Geotagging within OpenStack

Asset tags/geotags are made up of one or more user defined attributes, along with a way to make sure the tag is specifically assigned to an asset. Figure 18 depicts how an asset tag/geotag is composed.



Asset = any computing device,
Tag = 1 or more Attributes associated to the asset . Ex: Identity, location, price, customer, classification, ….)
Asset certificate – A digitally signed set of tags associated with an asset
Asset Tag – SHA-1 hash of the asset certificate

Figure 18: Composition of an Asset Tag/Geotag

In order for asset tags/geotags to be utilized in the OpenStack environment, there must be modifications made to the out-of-the-box OpenStack implementation. Figure 19 depicts what these changes are, and where in the OpenStack architecture they exist.

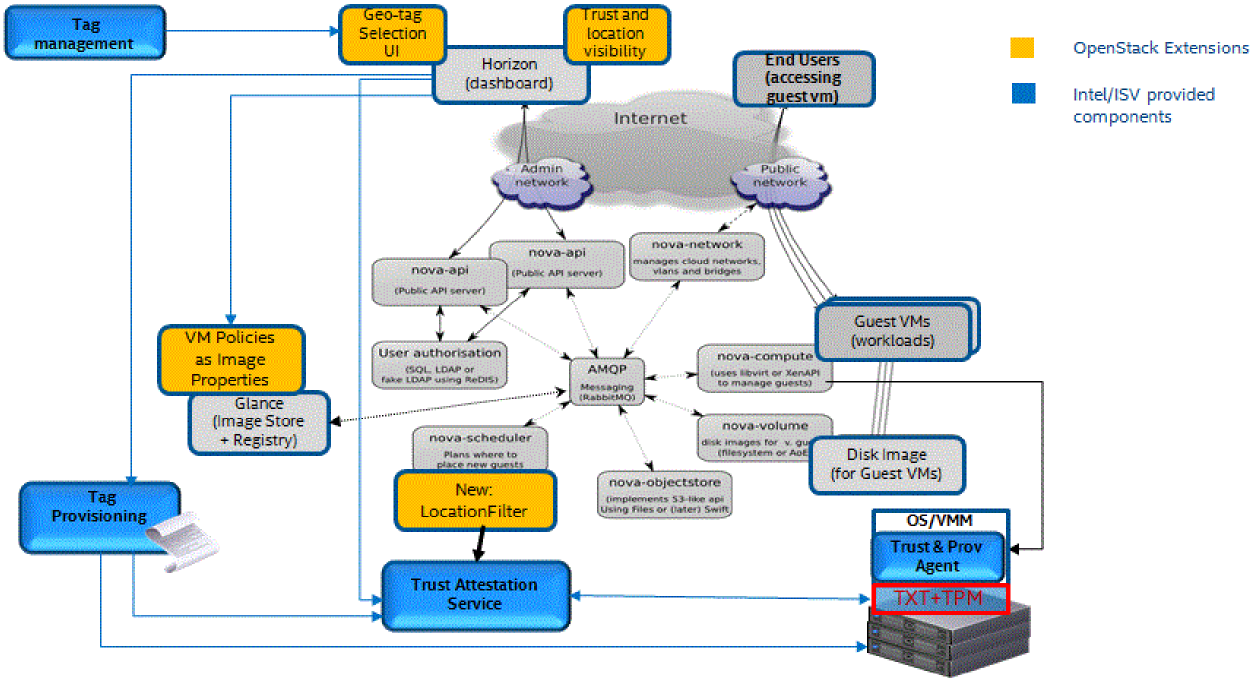

Tag management
Geo-tag Selection UI
Trust and location visibility
Horizon (dashboard)
End Users (accessing guest vm)
Internet
Admin network
Public network
🟨 OpenStack Extensions
🟦 Intel/ISV provided components
nova-api
(Public API server)
nova-api
(Public API server)
nova-network
manages cloud networks, vlans and bridges
Guest VMs (workloads)
VM Policies as Image Properties
User authorisation
(SQL, LDAP or fake LDAP using ReDIS)
AMQP
Messaging (RabbitMQ)
nova-compute
(uses libvirt or XenAPI to manage guests)
Glance (Image Store + Registry)
nova-volume
disk images for v. gu (filesystem or AoE
nova-scheduler
Plans where to place new guests
Disk Image (for Guest VMs)
Tag Provisioning
New: LocationFilter
nova-objectstore
(implements S3-like api Using Files or (later) Swift
OS/VMM
Trust & Prov Agent
TXT+TPM
Trust Attestation Service

Figure 19: Proposed OpenStack Changes

C.1 Solution Architecture

Figure 20 shows the architecture depicted in Appendix A, but with the specific products used in the OpenStack platform implementation.

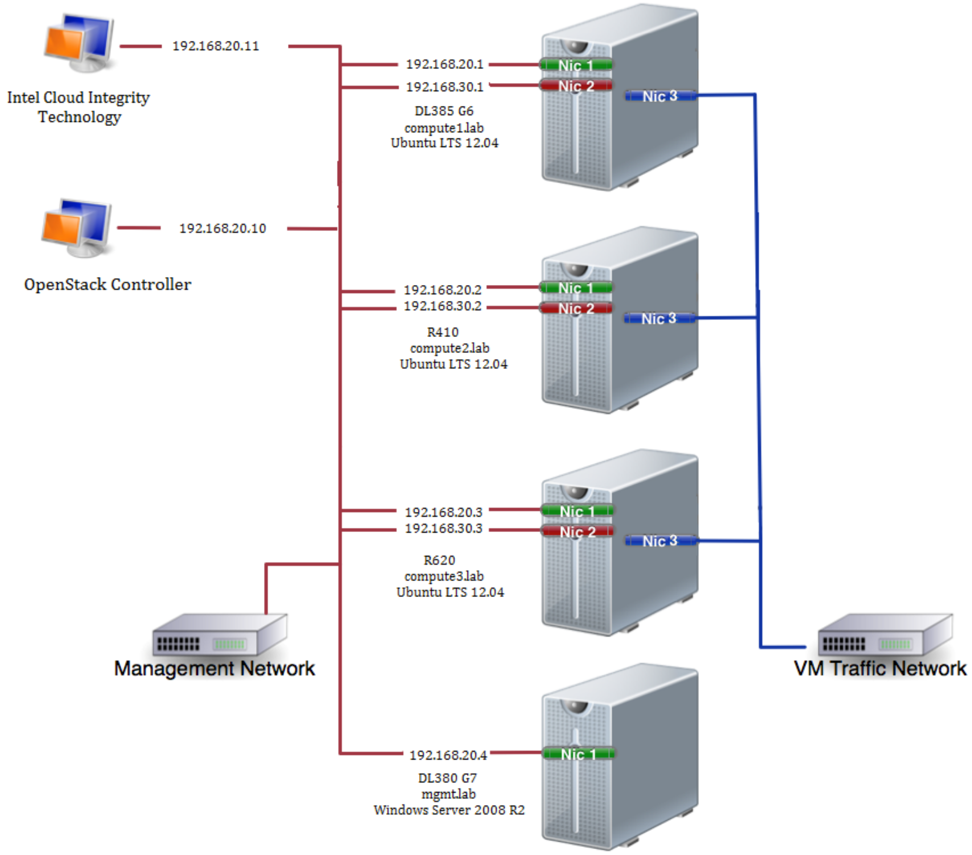


Figure 20: Proof of Concept Implementation

C.2 Hardware Description

The implemented architecture is composed of three Dell servers running Ubuntu 12.04 LTS with kernel-based virtual machine (KVM) configured as a cluster with a shared resource pool and a management node that includes two VMs providing different functionalities, and a dedicated management workstation.

Trusted Cloud Cluster:

- 1 x Dell PowerEdge R620 (Intel TXT enabled):
  - 2 x Intel Xeon Processor E5-2660 @ 2.20 GHz
  - 64 GB memory
  - Ubuntu 12.04 LTS
- 1 x Dell PowerEdge R410 (Intel TXT enabled):
  - 2 x Intel Xeon Processor E5630 @ 2.53 GHz
  - 8 GB
  - Ubuntu 12.04 LTS
- 1 x HP Proliant DL385 G6
  - Ubuntu 12.04 LTS

Management Node:

- HP Proliant DL380 G7
  - 2 x Intel Xeon Processor E5640 @ 2.67 GHz
  - 12 GB Memory
- Windows Server 2008 R2 Hyper-V hosting the following VMs:
  - Ubuntu 12.04 LTS with OpenStack IceHouse Controller
  - Ubuntu 12.04 LTS with Intel Cloud Integrity Technology appliance

C.3 BIOS Changes

The following changes are required in the BIOS settings:

1. Set Intel TXT to “Enabled”.
2. Set Intel Virtualization Technology (Intel VT) to “Enabled”.
3. Set Intel VT for Directed I/O to “Enabled”.
4. Set “Administrator Password” and reboot prior to enabling the TPM.
5. Change TPM Administrative Control to “Turn On”; TPM state will show as “enabled and activated” after reboot.

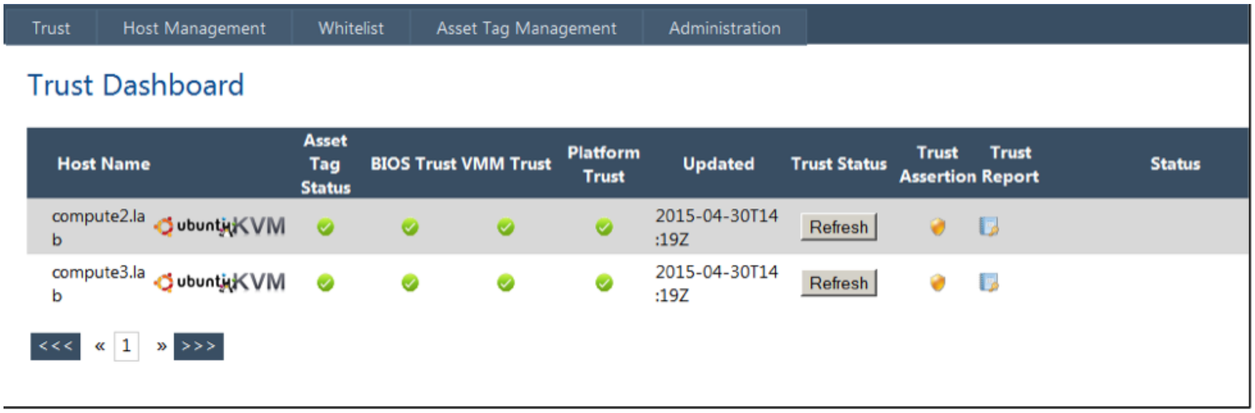
C.4 OpenStack Components

This implementation is running a base install of OpenStack Icehouse, with installation steps followed from the OpenStack official documentation (found at http://docs.openstack.org/icehouse/install-guide/install/apt/content/). The base install includes a single controller node running the identity service (Keystone), the image service (Glance), the networking service, the compute service (Nova), and the dashboard (Horizon), with each additional compute node running the compute service (Nova). Each OpenStack node is running on Ubuntu 12.04 LTS with all of the compute nodes running on its own physical machine while the controller is running within a VM.

C.5 Trust Authority Installation, Configuration, and Registration

The Trust Authority comes as an Intel virtual appliance, Intel Cloud Integrity Technology, which is an easily deployable VM with an install script and answer file for installation of the required services. The Intel Cloud Integrity Technology virtual appliance is available on the Intel FTP site. Instructions for completing the answer file and running the Intel Cloud Integrity Technology installer can be also found in Intel’s FTP site as the Intel Cloud Integrity Technology Product Guide. As part of the Intel Cloud Integrity Technology package, there is also a trust agent that must be installed on each compute node that is TXT and TPM enabled. This trust agent will allow the compute node to register and attest to the Intel Cloud Integrity Technology server, as well as act as the mechanism for Intel Cloud Integrity Technology to push the geotags to each compute node. The Intel trust agent is installed via an install script and answer file, both of which are found on the Intel FTP site along with documentation on how to populate the answer file and run the install script.

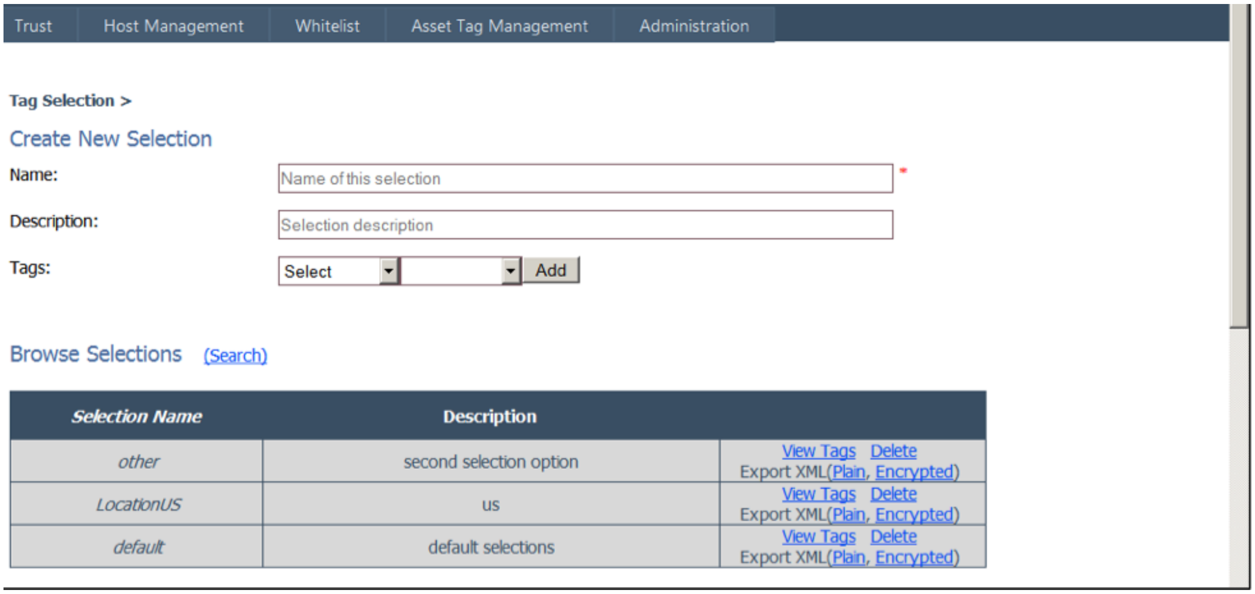
Once the Intel Cloud Integrity Technology server is installed, along with the trust agents on the compute nodes, each compute node can be registered into the Intel Cloud Integrity Technology. This is done by through the “Host Management” tab in the Intel Cloud Integrity Technology URL. Each host is imported by its IP address or hostname; once they are imported into the Intel Cloud Integrity Technology appliance, the trust status of each will be visible in the Intel Cloud Integrity Technology Trust Dashboard, as shown in Figure 21.



Host Name		Asset Tag Status	BIOS Trust	VMM Trust	Platform Trust	Updated	Trust Status	Trust Assertion	Trust Report	Status
compute2.lab	ubuntu KVM	✓	✓	✓	✓	2015-04-30T14:19Z	Refresh			
compute3.lab	ubuntu KVM	✓	✓	✓	✓	2015-04-30T14:19Z	Refresh			

Figure 21: Intel Cloud Integrity Technology Trust Dashboard

Through the "Asset Tag Management" tab in the Intel Cloud Integrity Technology URL, geotags can be created to be pushed to each node that is registered with Intel Cloud Integrity Technology. Figure 22 shows what the Asset Tag Management page looks like, as well as its functionality on how to create new tag values.



Name: Name of this selection *

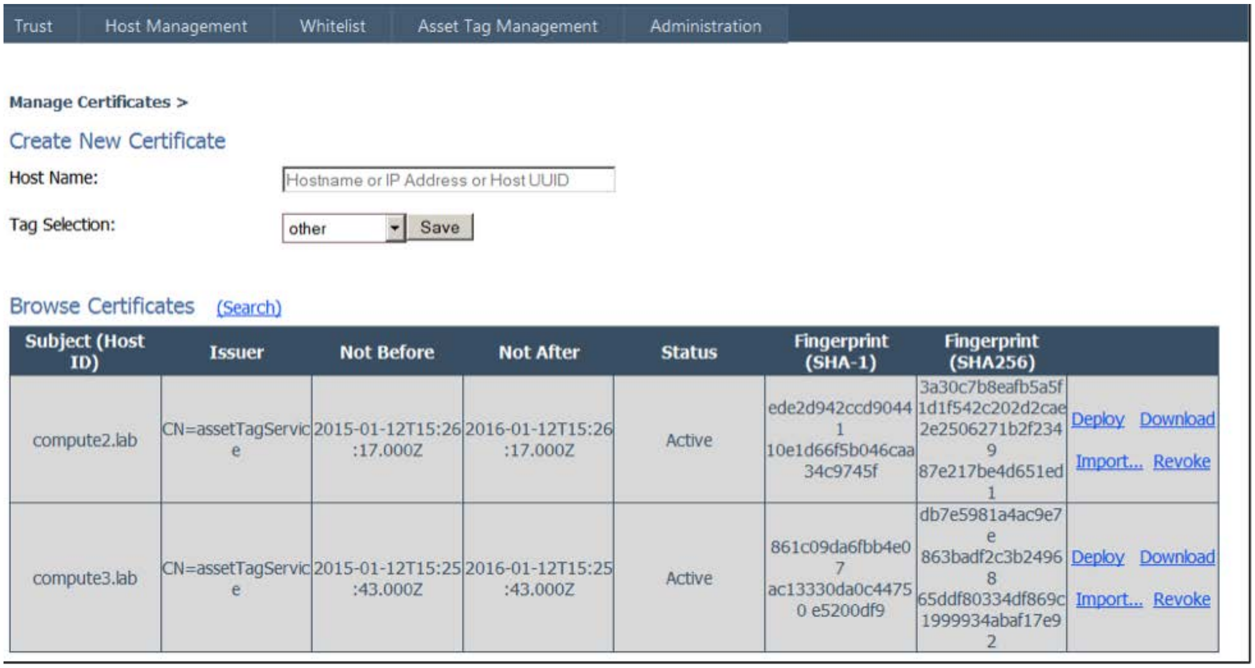
Description: Selection description

Tags: Select

Selection Name	Description	
other	second selection option	View Tags Delete Export XML(Plain, Encrypted)
LocationUS	us	View Tags Delete Export XML(Plain, Encrypted)
default	default selections	View Tags Delete Export XML(Plain, Encrypted)

Figure 22: Intel Cloud Integrity Technology Tag Management

Once geotags are created for the compute nodes, through the "Asset Tag Management" → "Manage Certificates" tab, geotags can be pushed to each compute node. Figure 23 depicts which certificates have been provisioned to hosts, and also the mechanisms to deploy new certificates or revoke current certificates.



Create New Certificate

Host Name: Hostname or IP Address or Host UUID

Tag Selection: other

Browse Certificates

Subject (Host ID)	Issuer	Not Before	Not After	Status	Fingerprint (SHA-1)	Fingerprint (SHA256)	
compute2.lab	CN=assetTagService	2015-01-12T15:26:17.000Z	2016-01-12T15:26:17.000Z	Active	ede2d942ccd90441 10e1d66f5b046caa 34c9745f	3a30c7b8eafb5a5f 1d1f542c202d2cae 2e2506271b2f2349 87e217be4d651ed1	Deploy Download Import... Revoke
compute3.lab	CN=assetTagService	2015-01-12T15:25:43.000Z	2016-01-12T15:25:43.000Z	Active	861c09da6fbb4e07 ac13330da0c44750 e5200df9	db7e5981a4ac9e7e 863badf2c3b24968 65ddf80334df869c 1999934abaf17e92	Deploy Download Import... Revoke

Figure 23: Intel Cloud Integrity Technology Certificate Management

Once geotags have been pushed to the compute nodes, OpenStack services can be modified to ensure that VM migration is enforced by policies that correspond to compute node trust and geotags.

C.6 Trust Authority and OpenStack Integration

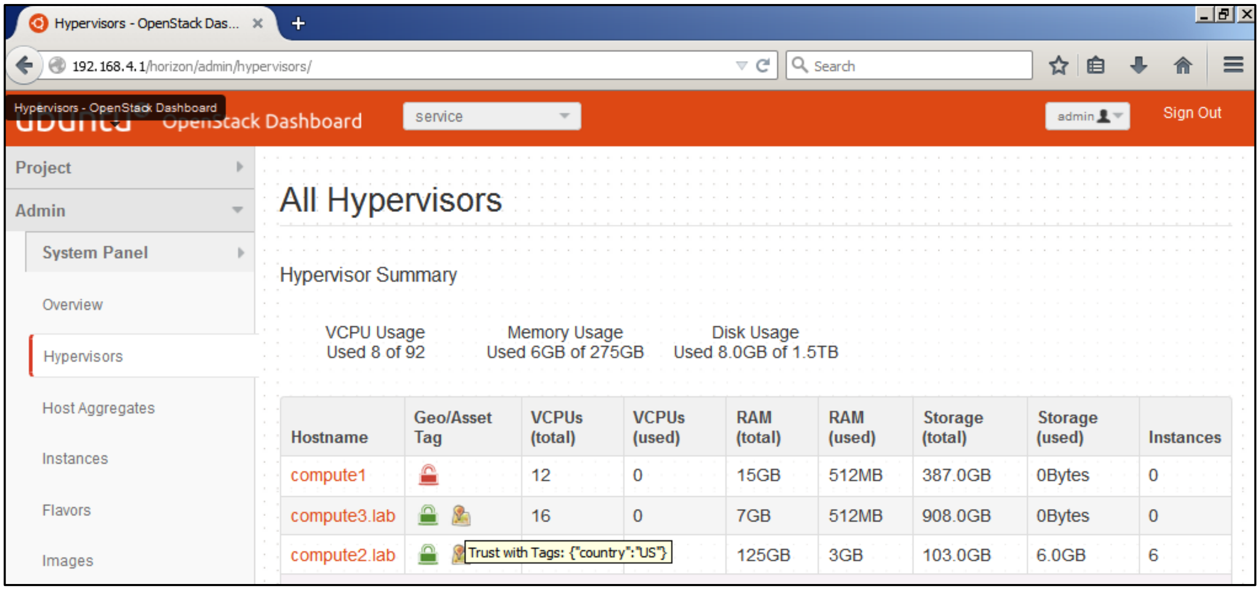
Before OpenStack can use the trust attestation that Intel Cloud Integrity Technology provides, it first must know how to communicate with the server, as well as understand how to use the trust assertions that Intel Cloud Integrity Technology provides. Since VM migration policies will be enforced based off the image that instances are launched from, the properties associated with the OpenStack images must be modified. Also, since instance creation is performed through the OpenStack Horizon dashboard, the dashboard code must be modified to reflect the trust policies that can now be associated with images and instances. Finally, to enforce policy-based VM migration, the Nova scheduler must be modified so that it can get the correct trust assertions from Intel Cloud Integrity Technology. Intel provides an OpenStack/Intel Cloud Integrity Technology integration package that automates the above OpenStack modifications. Once the package has been downloaded from the Intel FTP site, the following steps need to be taken:

1. Place the integration package in root’s home folder on the OpenStack controller
2. Make the install script executable – # chmod +x icehouse_geo-tag_patch.tgz
3. Extract the package – # tar xczf “icehouse_geo-tag_patch.tgz”
4. Go into the directory that has been create – # cd icehouse
5. Before applying the patch, update nova_settings and horizon_settings files to change attestation server IP and access credentials
6. Remove Ubuntu OpenStack Themes – # apt-get remove --purge openstack-dashboard-ubuntu-theme
7. Run the install script – # ./setup

The manual steps that the installer automates can be found in Intel documentation on the FTP server under OpenStack documentation. Also, the changes to the OpenStack components that have been made had blueprints submitted to the official OpenStack project (https://review.openstack.org/#/c/133106) as well as code changes for the OpenStack Nova filter (https://review.openstack.org/#/c/141214).

C.7 OpenStack Usage

After the OpenStack and Intel Cloud Integrity Technology installation and integration have been completed, it is time to create OpenStack instances that will have migration policies based on Intel Cloud Integrity Technology trust attestations. The first step is to log into the OpenStack Horizon dashboard and under the Admin panel, select Hypervisors. Here all of the compute nodes that are registered with the OpenStack controller will be listed. Figure 24 shows these compute nodes along with the extension for Geo/Asset Tag in the hypervisor dashboard.


Hypervisors - OpenStack Das...
192.168.4.1/horizon/admin/hypervisors/
Search
Hypervisors - OpenStack Dashboard
ubuntu OpenStack Dashboard
service
admin
Sign Out
Project
Admin
System Panel
Overview
Hypervisors
Host Aggregates
Instances
Flavors
Images

All Hypervisors

Hypervisor Summary

VCPU Usage
Used 8 of 92

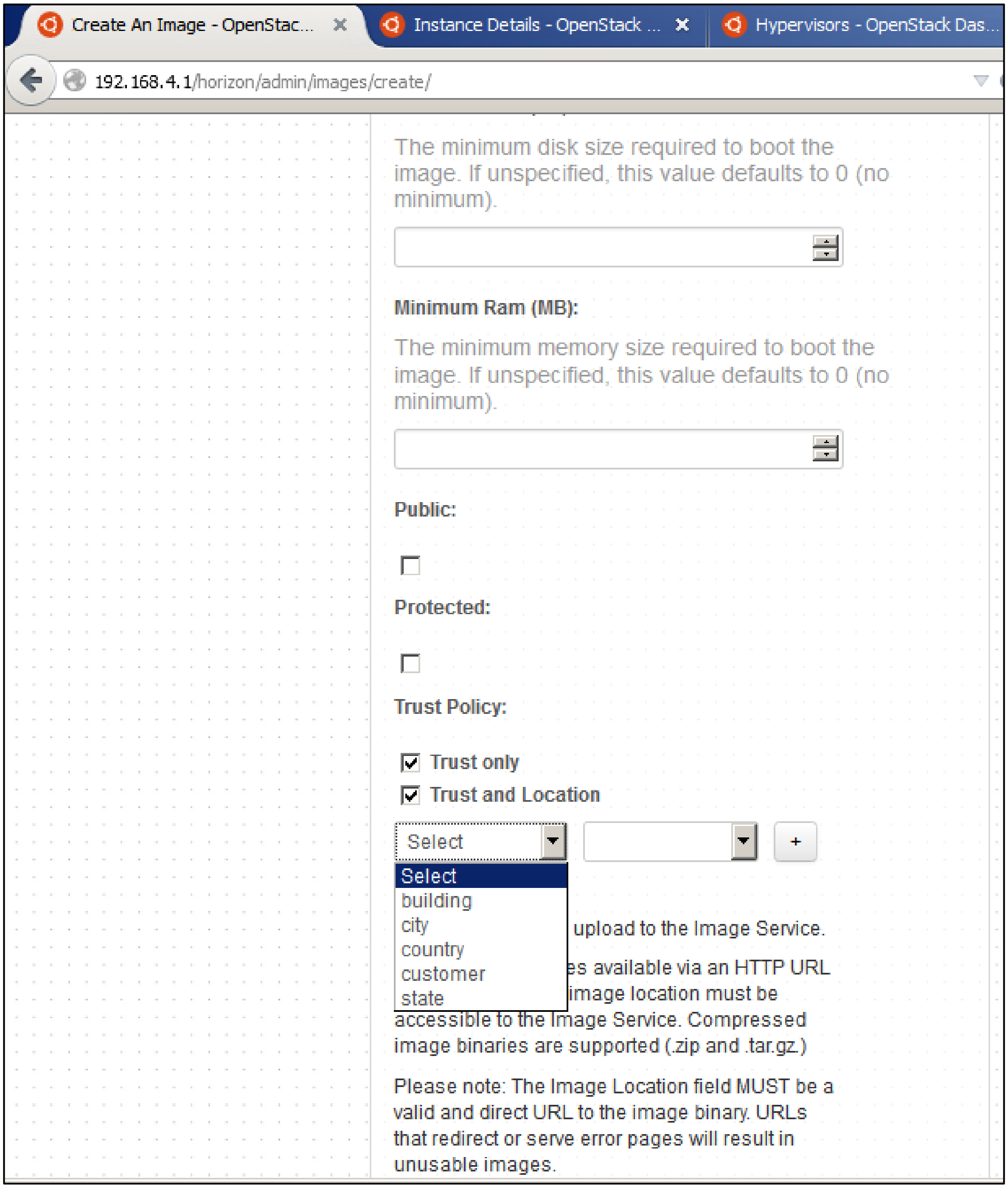
Memory Usage
Used 6GB of 275GB

Disk Usage
Used 8.0GB of 1.5TB

Hostname	Geo/Asset Tag	VCPUs (total)	VCPUs (used)	RAM (total)	RAM (used)	Storage (total)	Storage (used)	Instances
compute1	🔓	12	0	15GB	512MB	387.0GB	0Bytes	0
compute3.lab	🔒 📍	16	0	7GB	512MB	908.0GB	0Bytes	0
compute2.lab	🔒 📍 Trust with Tags: {"country":"US"}			125GB	3GB	103.0GB	6.0GB	6

Figure 24: OpenStack Hypervisor Dashboard

Notice that compute2.lab and compute3.lab have the Trusted Boot and Trusted Geolocation icons, which is representative of what was seen in the Intel Cloud Integrity Technology dashboard. The next step is to create an OpenStack image that will leverage these trust attestations. To do so, under the Admin panel choose the Images selection and click the button to Create an image. Figure 25 shows the options that will appear to apply trust policies to the image that will be created.


Create An Image - OpenStac...
Instance Details - OpenStack ...
Hypervisors - OpenStack Das...
192.168.4.1/horizon/admin/images/create/
Select
building
city
country
customer
state
+

The minimum disk size required to boot the image. If unspecified, this value defaults to 0 (no minimum).

Minimum Ram (MB):

The minimum memory size required to boot the image. If unspecified, this value defaults to 0 (no minimum).

Public:

□

Protected:

□

Trust Policy:

☑ Trust only

☑ Trust and Location

upload to the Image Service.

es available via an HTTP URL image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

Figure 25: OpenStack Image Creation with Trust Policies

The options exist to apply no trust policies, to apply a policy that only Trusted Boot is required, or to require Trusted Boot and Trusted Geolocation for each instance that will be launched from this image. In the reference implementation, one image for each condition has been created. Figure 26 shows the images that have been created along with the trust policies that have been applied to them.

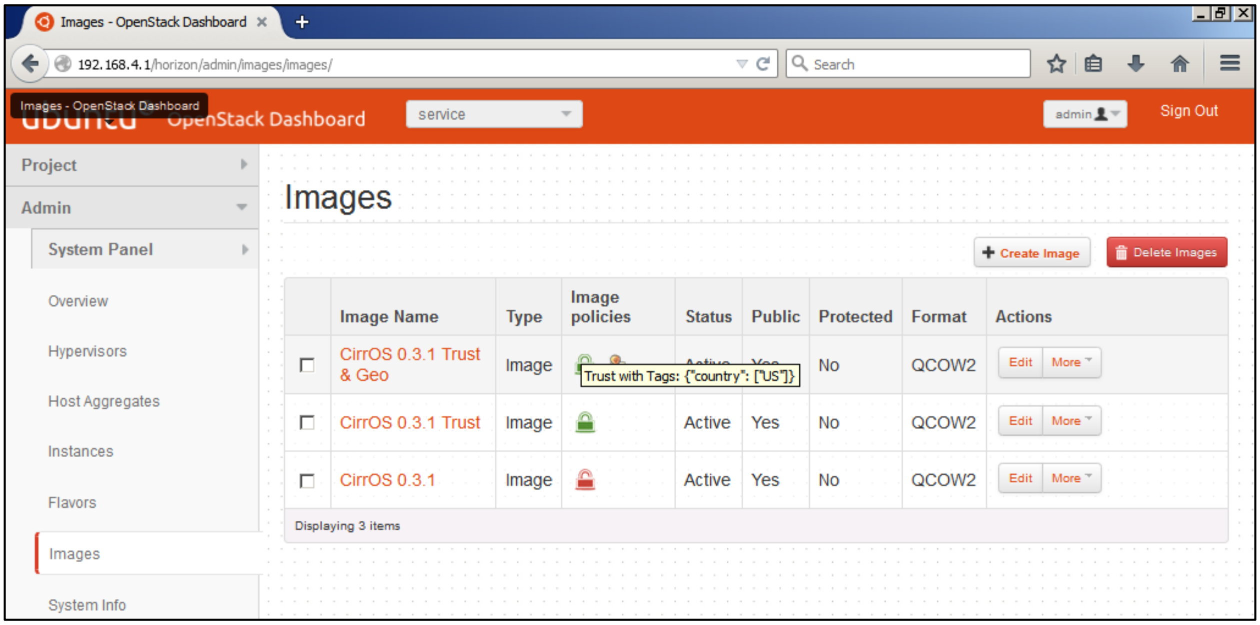
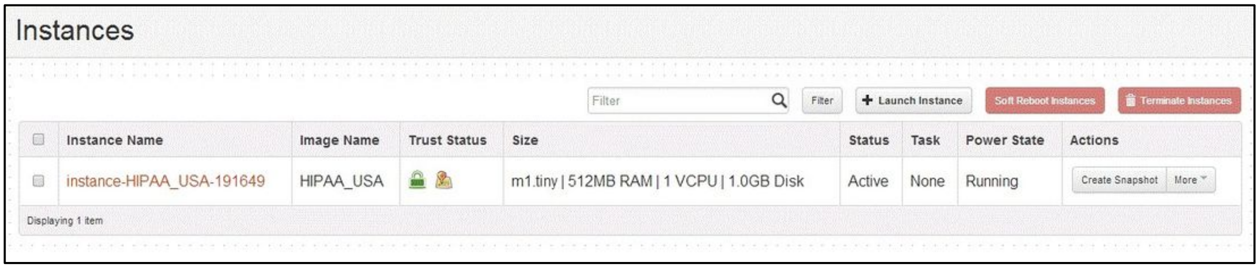


	Image Name	Type	Image policies	Status	Public	Protected	Format	Actions
□	CirrOS 0.3.1 Trust & Geo	Image	🔒 Trust with Tags: {"country": ["US"]}	Active	Yes	No	QCOW2	Edit More
□	CirrOS 0.3.1 Trust	Image	🔒	Active	Yes	No	QCOW2	Edit More
□	CirrOS 0.3.1	Image	🔓	Active	Yes	No	QCOW2	Edit More

Displaying 3 items

Figure 26: OpenStack Images Dashboard

When an instance is launched from a specific image, the instance will inherit the trust policies from the image. Figure 27 depicts a running instance with Trusted Boot and Trusted Geolocation policies.



□	Instance Name	Image Name	Trust Status	Size	Status	Task	Power State	Actions
□	instance-HIPAA_USA-191649	HIPAA_USA	🔒	m1.tiny | 512MB RAM | 1 VCPU | 1.0GB Disk	Active	None	Running	Create Snapshot More

Displaying 1 item

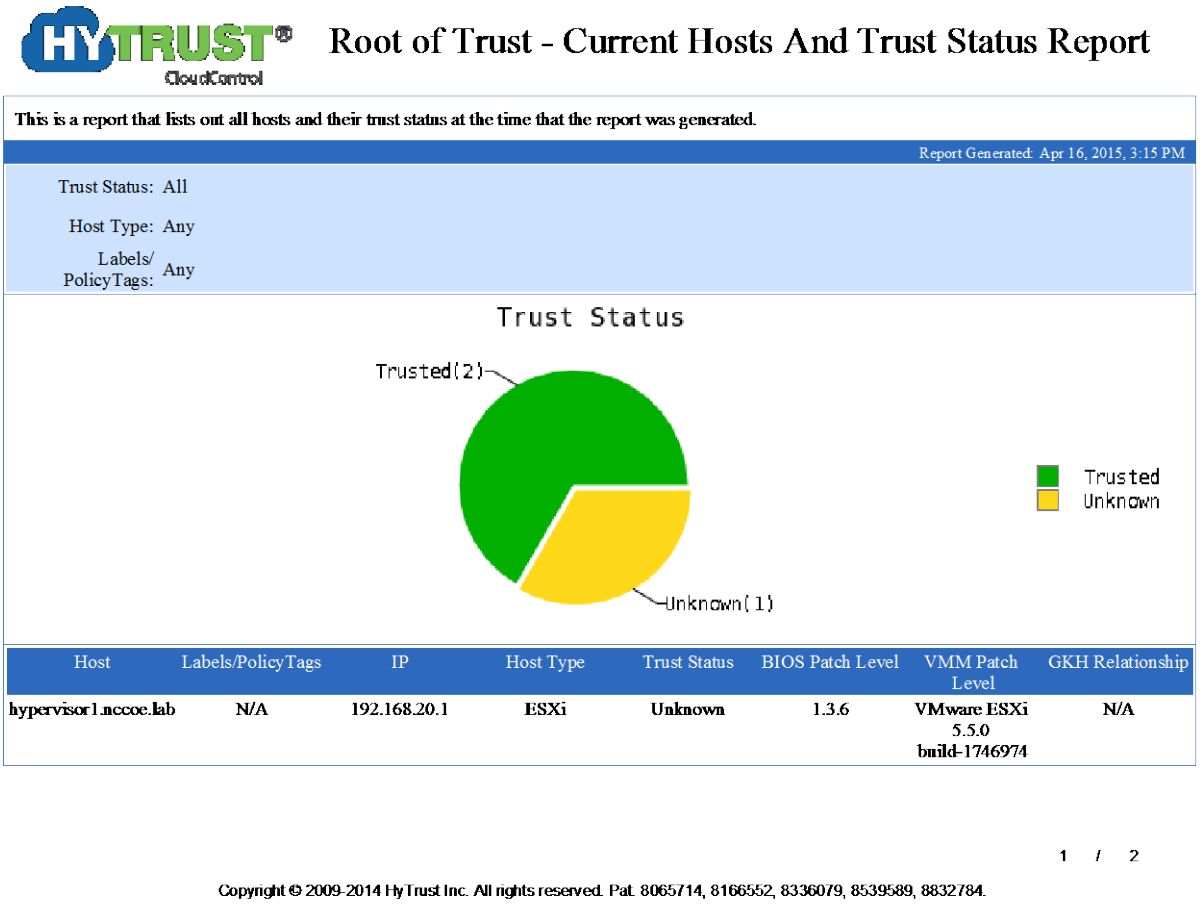
Figure 27: OpenStack Instance Dashboard

For example, when an instance is launched from "CirrOS 0.3.1 Trust & Geo", the Nova scheduler will initially place the VM instance on a compute node that meets the Trusted Boot and Trusted Geolocation policies. Furthermore, when a migration on the VM is requested, the Nova scheduler will attempt to find another compute node that matches the trust policies. If such a compute node is found then the Nova scheduler will start migration to that host; however, if no compute node matching the trust policy requirements is found then the Nova scheduler will not perform a migration of the VM instance.

Appendix D—Reporting Implementation: HyTrust

This appendix presents screen shots from the HTCC product that demonstrate the monitoring of measurements in a governance, risk, and compliance dashboard.

Figures 28 and 29 show a chart reflecting the relative size of the pools of trusted (green) and unknown/untrusted (yellow) cloud servers. In this example, there are two servers in the trusted pool and one server in the untrusted pool. Relevant information for each server is provided: the hostname, applicable labels and policy tags, IP address, type of host, trust status, BIOS level, hypervisor patch level, and relationship to a trusted good known host.



HYTRUST® CloudControl

Root of Trust - Current Hosts And Trust Status Report

This is a report that lists out all hosts and their trust status at the time that the report was generated.

Report Generated: Apr 16, 2015, 3:15 PM

Trust Status: All

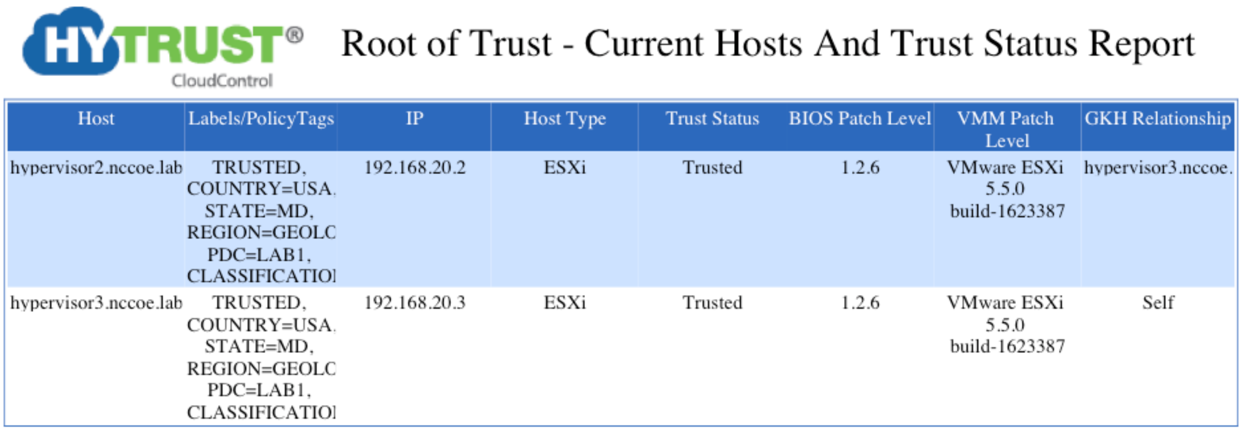
Host Type: Any

Labels/ PolicyTags: Any

Host	Labels/PolicyTags	IP	Host Type	Trust Status	BIOS Patch Level	VMM Patch Level	GKH Relationship
hypervisor1.nccoe.lab	N/A	192.168.20.1	ESXi	Unknown	1.3.6	VMware ESXi 5.5.0 build-1746974	N/A

1 / 2

Figure 28: HyTrust Report Page 1 of 2



Root of Trust - Current Hosts And Trust Status Report

Host	Labels/PolicyTags	IP	Host Type	Trust Status	BIOS Patch Level	VMM Patch Level	GKH Relationship
hypervisor2.nccoe.lab	TRUSTED, COUNTRY=USA, STATE=MD, REGION=GEOLC PDC=LAB1, CLASSIFICATIOI	192.168.20.2	ESXi	Trusted	1.2.6	VMware ESXi 5.5.0 build-1623387	hypervisor3.nccoe.
hypervisor3.nccoe.lab	TRUSTED, COUNTRY=USA, STATE=MD, REGION=GEOLC PDC=LAB1, CLASSIFICATIOI	192.168.20.3	ESXi	Trusted	1.2.6	VMware ESXi 5.5.0 build-1623387	Self

Figure 29: HyTrust Report Page 2 of 2

To create this specific report, perform the following steps:

1. Enable Reports: General > Reports > Check Enable (No need for email) - On page 167 in the Admin Guide http://downloads.hytrust.com/product_documentation/4.1.0/HyTrust_CloudControl_Administration_Guide.pdf
2. Add > Root of Trust – Current Hosts and Trust Status Report > Name: Current_Hosts_and_Trust_Status_Report - On page 183 in the Admin Guide http://downloads.hytrust.com/product_documentation/4.1.0/HyTrust_CloudControl_Administration_Guide.pdf
3. Click > Apply
4. Click > PDF (It will download a PDF and then you can open it)

Custom reports can be made and exported through the HTCC web interface. This is done at the General > Reports tab. For more detailed information on how to create custom reports, refer to the HyTrust Administration Guide.

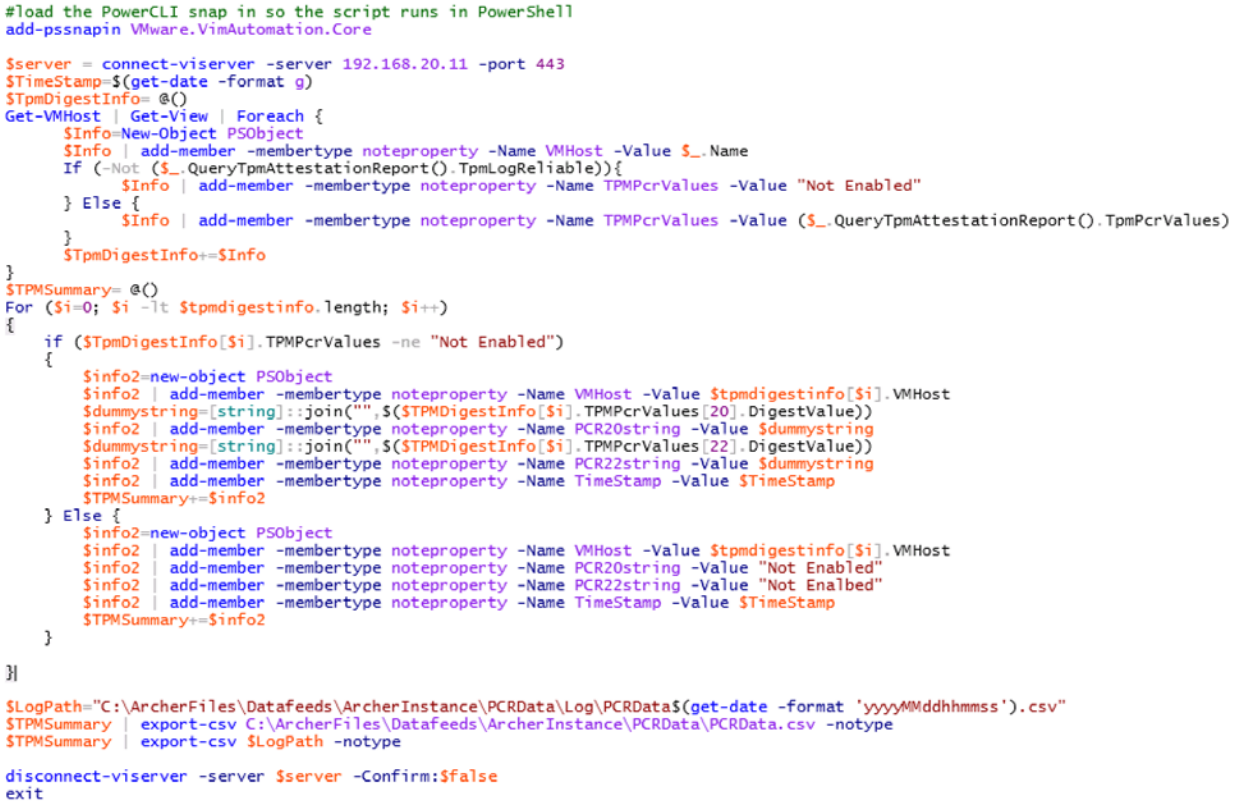
Appendix E—Reporting Implementation: RSA Archer

This appendix presents screen shots from the RSA Archer product that demonstrate the monitoring of measurements in a governance, risk, and compliance dashboard.

Successful implementation of the RSA Archer GRC Solution as a part of this proof of concept involved an install of the base, out-of-the-box Archer GRC Platform Version 5.5 SP2. Leveraging two Archer On-Demand Applications, the Archer Platform is primed to consume the Intel/VMWare PCR data values and display dashboard data. The Archer GRC Platform’s open integration components enabled configuration of a script to authenticate to the VMWare VSphere Infrastructure, collect the PCR data values, and populate the Archer Application.

In order to collect this data, Archer leveraged Microsoft’s PowerShell 3.0 and VMWare’s PowerCLI 6.0 to programmatically connect, authenticate, and read the PCR data values from the established Intel/VMWare infrastructure. These values, continually collected, are logged into the Archer On-Demand Application with appropriate timestamp and association to the related systems. Once in the Archer Platform, dashboard displays are immediately updated to provide an easy-to-understand red/green status for secure boot and geolocation policy compliance. In addition, Archer-configured active notifications are immediately distributed to systems or individuals that need to respond to a change in the geolocation data.

Figure 30 shows the PowerShell script that reads the PCR data from the vCenter Server.



```
#load the PowerCLI snap in so the script runs in PowerShell
add-pssnapin VMware.VimAutomation.Core

$server = connect-viserver -server 192.168.20.11 -port 443
$TimeStamp=$(get-date -format g)
$TpmDigestInfo= @()
Get-VMHost | Get-View | Foreach {
      $Info=New-Object PSObject
      $Info | add-member -membertype noteproperty -Name VMHost -Value $_.Name
      If (-Not ($_.QueryTpmAttestationReport().TpmLogReliable)){
            $Info | add-member -membertype noteproperty -Name TPMPcrValues -Value "Not Enabled"
      } Else {
            $Info | add-member -membertype noteproperty -Name TPMPcrValues -Value ($_.QueryTpmAttestationReport().TpmPcrValues)
      }
      $TpmDigestInfo+=$Info
}
$TPMSummary= @()
For ($i=0; $i -lt $tpmdigestinfo.length; $i++)
{
    if ($TpmDigestInfo[$i].TPMPcrValues -ne "Not Enabled")
    {
        $info2=new-object PSObject
        $info2 | add-member -membertype noteproperty -Name VMHost -Value $tpmdigestinfo[$i].VMHost
        $dummystring=[string]::join("",$($TPMDigestInfo[$i].TPMPcrValues[20].DigestValue))
        $info2 | add-member -membertype noteproperty -Name PCR20string -Value $dummystring
        $dummystring=[string]::join("",$($TPMDigestInfo[$i].TPMPcrValues[22].DigestValue))
        $info2 | add-member -membertype noteproperty -Name PCR22string -Value $dummystring
        $info2 | add-member -membertype noteproperty -Name TimeStamp -Value $TimeStamp
        $TPMSummary+=$info2
    } Else {
        $info2=new-object PSObject
        $info2 | add-member -membertype noteproperty -Name VMHost -Value $tpmdigestinfo[$i].VMHost
        $info2 | add-member -membertype noteproperty -Name PCR20string -Value "Not Enabled"
        $info2 | add-member -membertype noteproperty -Name PCR22string -Value "Not Enalbed"
        $info2 | add-member -membertype noteproperty -Name TimeStamp -Value $TimeStamp
        $TPMSummary+=$info2
    }

}

$LogPath="C:\ArcherFiles\Datafeeds\ArcherInstance\PCRData\Log\PCRData$(get-date -format 'yyyyMMddhhmmss').csv"
$TPMSummary | export-csv C:\ArcherFiles\Datafeeds\ArcherInstance\PCRData\PCRData.csv -notype
$TPMSummary | export-csv $LogPath -notype

disconnect-viserver -server $server -Confirm:$false
exit
```

Figure 30: PowerShell Script

A Windows Scheduled Task is created to run this script every ten minutes to continually update the PCR data file. Once these values are written to the PCRData.csv file, Archer uses its data import functionality feed to populate a custom field with the values. The Archer data import is also set on a schedule to continually run and populate Archer with the most current PCR data. Figure 31 shows the Archer default dashboard view for overall system trust.

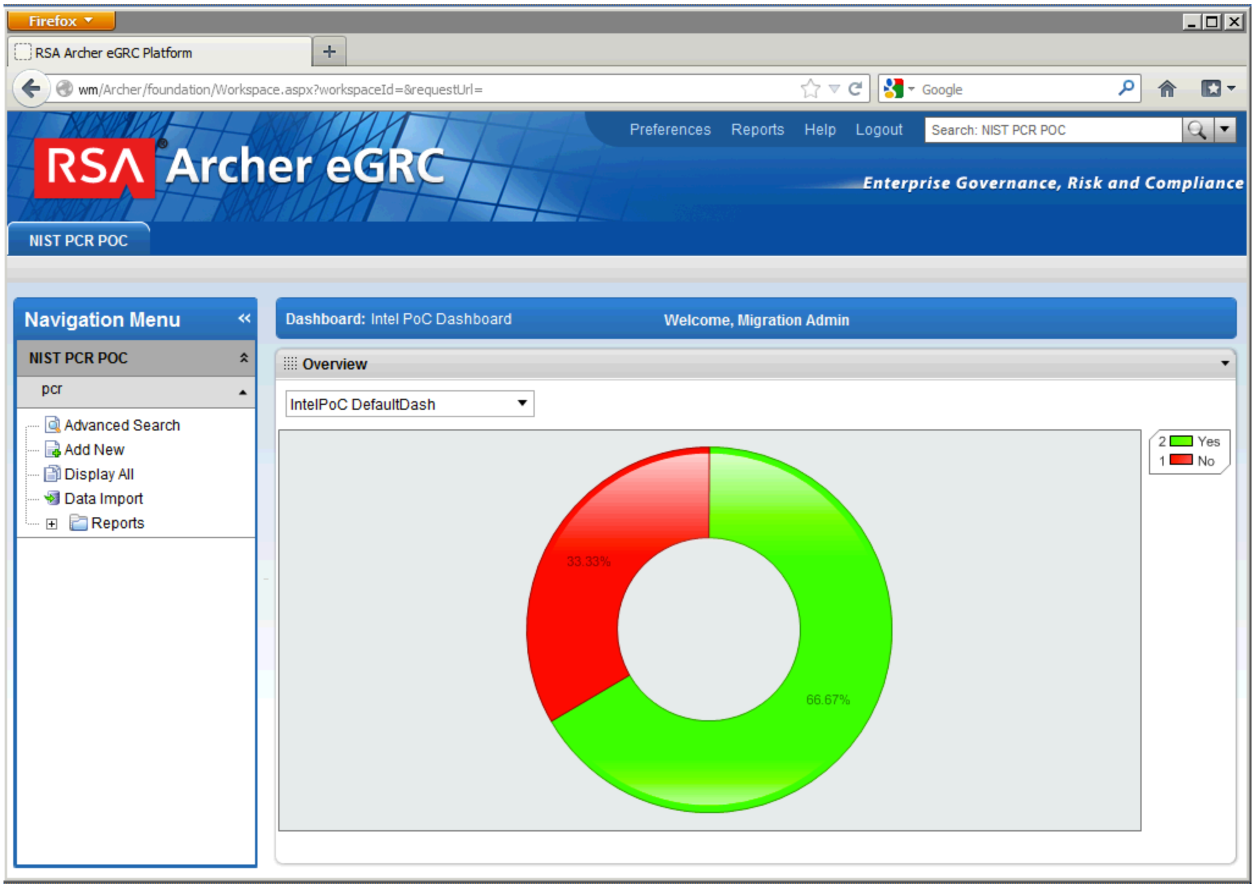
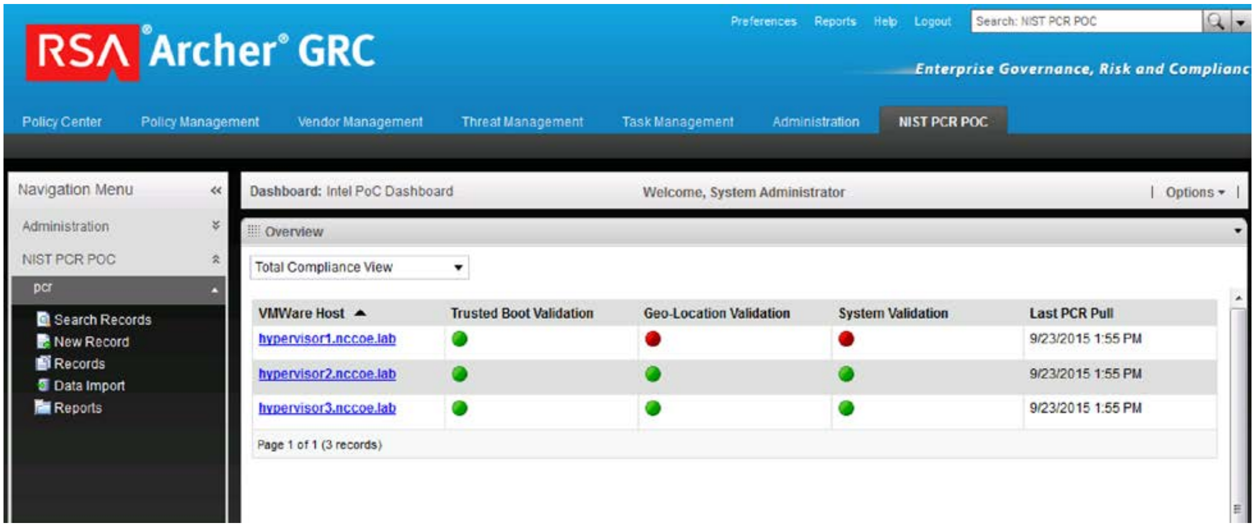


Figure 31: Archer Default Dashboard

Figure 32 shows each host's trust status for BIOS/VMM and geolocation, as well as the last time the values were updated.



Dashboard: Intel PoC Dashboard

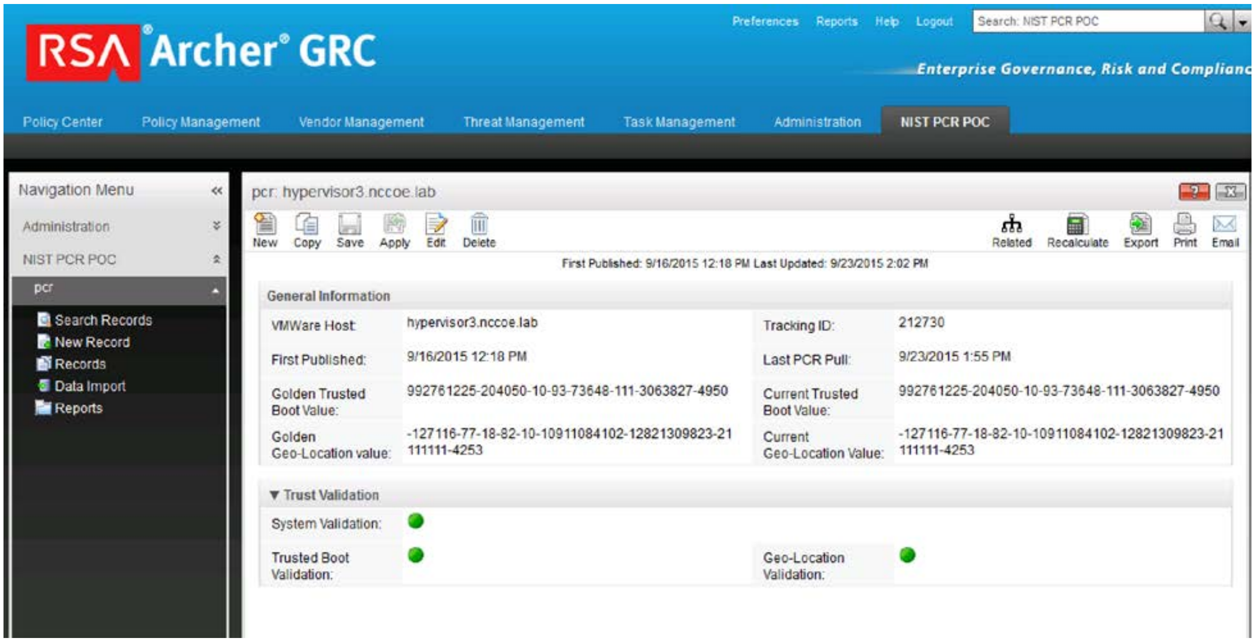
Welcome, System Administrator

VMWare Host	Trusted Boot Validation	Geo-Location Validation	System Validation	Last PCR Pull
hypervisor1.nccoe.lab	🟢	🔴	🔴	9/23/2015 1:55 PM
hypervisor2.nccoe.lab	🟢	🟢	🟢	9/23/2015 1:55 PM
hypervisor3.nccoe.lab	🟢	🟢	🟢	9/23/2015 1:55 PM

Page 1 of 1 (3 records)

Figure 32: Host Trust Status

Figure 33 provides a detailed view for one specific host. It allows the user to see the trusted boot and geolocation values, current boot and geolocation values, provisioning time, and last PCR data update for the host.



pcr: hypervisor3.nccoe.lab

First Published: 9/16/2015 12:18 PM Last Updated: 9/23/2015 2:02 PM

General Information

VMWare Host:	hypervisor3.nccoe.lab	Tracking ID:	212730
First Published:	9/16/2015 12:18 PM	Last PCR Pull:	9/23/2015 1:55 PM
Golden Trusted Boot Value:	992761225-204050-10-93-73648-111-3063827-4950	Current Trusted Boot Value:	992761225-204050-10-93-73648-111-3063827-4950
Golden Geo-Location value:	-127116-77-18-82-10-10911084102-12821309823-21 111111-4253	Current Geo-Location Value:	-127116-77-18-82-10-10911084102-12821309823-21 111111-4253

▼ Trust Validation

System Validation:	🟢		
Trusted Boot Validation:	🟢	Geo-Location Validation:	🟢

Figure 33: Detailed View of a Host

Appendix F—Supporting NIST SP 800-53 Security Controls and Publications

The major controls in the NIST Special Publication 800-53 Revision 4, *Security and Privacy Controls for Federal Information Systems and Organizations* control catalog that affect the trusted geolocation proof of concept implementation are:

AU-2, Audit Events

Related controls: AC-6, AC-17, AU-3, AU-12, MA-4, MP-2, MP-4, SI-4

References: NIST Special Publication 800-92; Web: csrc.nist.gov/pcig/cig.html, idmanagement.gov

CA-2, Security Assessments

Related controls: CA-5, CA-6, CA-7, PM-9, RA-5, SA-11, SA-12, SI-4

References: Executive Order 13587; FIPS Publication 199; NIST Special Publications 800-37, 800-39, 800-53A, 800-115, 800-137

CA-7, **Continuous Monitoring**

Related controls: CA-2, CA-5, CA-6, CM-3, CM-4, PM-6, PM-9, RA-5, SA-11, SA-12, SI-2, SI-4

References: OMB Memorandum 11-33; NIST Special Publications 800-37, 800-39, 800-53A, 800-115, 800-137; US-CERT Technical Cyber Security Alerts; DoD Information Assurance Vulnerability Alerts

CM-2, Baseline Configuration

Related controls: CM-3, CM-6, CM-8, CM-9, SA-10, PM-5, PM-7

References: NIST Special Publication 800-128

CM-3, Configuration Change Control

Related controls: CM-2, CM-4, CM-5, CM-6, CM-9, SA-10, SI-2, SI-12

References: NIST Special Publication 800-128

CM-8, Information System Component Inventory

Related controls: CM-2, CM-6, PM-5

References: NIST Special Publication 800-128

SC-2, Application Partitioning

Related controls: SA-4, SA-8, SC-3

SC-4, Information in Shared Resources

Related controls: AC-3, AC-4, MP-6

SC-7, Boundary Protection

Related controls: AC-4, AC-17, CA-3, CM-7, CP-8, IR-4, RA-3, SC-5, SC-13

References: FIPS Publication 199; NIST Special Publications 800-41, 800-77

SC-11, Trusted Path

Related controls: AC-16, AC-25

SC-29, Heterogeneity

Related controls: SA-12, SA-14, SC-27

SC-32, Information System Partitioning

Related controls: AC-4, SA-8, SC-2, SC-3, SC-7

References: FIPS Publication 199

SI-3, Malicious Code Protection

Related controls: CM-3, MP-2, SA-4, SA-8, SA-12, SA-13, SC-7, SC-26, SC-44, SI-2, SI-4, SI-7

References: NIST Special Publication 800-83

SI-4, **Information System Monitoring**

Related controls: AC-3, AC-4, AC-8, AC-17, AU-2, AU-6, AU-7, AU-9, AU-12, CA-7, IR-4, PE-3, RA-5, SC-7, SC-26, SC-35, SI-3, SI-7

References: NIST Special Publications 800-61, 800-83, 800-92, 800-94, 800-137

SI-6, Security Function Verification

Related controls: CA-7, CM-6

SI-7, Software, Firmware, and Information Integrity

Related controls: SA-12, SC-8, SC-13, SI-3

References: NIST Special Publications 800-147, 800-155

Information on these controls and guidelines on possible implementations can be found in the following publications:

- *SP 800-37 Rev. 1, Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach*
- *SP 800-39, Managing Information Security Risk: Organization, Mission, and Information System View*
- *SP 800-41 Rev. 1, Guidelines on Firewalls and Firewall Policy*
- *SP 800-53 Rev. 4, Security and Privacy Controls for Federal Information Systems and Organizations*
- *SP 800-53A Rev. 4, Assessing Security and Privacy Controls in Federal Information Systems and Organizations*
- *SP 800-61 Rev. 2, Computer Security Incident Handling Guide*
- *SP 800-77, Guide to IPsec VPNs*
- *SP 800-83 Rev. 1, Guide to Malware Incident Prevention and Handling for Desktops and Laptops*
- *SP 800-92, Guide to Computer Security Log Management*
- *Draft SP 800-94 Rev. 1, Guide to Intrusion Detection and Prevention Systems (IDPS)*
- *SP 800-100, Information Security Handbook: A Guide for Managers*
- *SP 800-115, Technical Guide to Information Security Testing and Assessment*
- *SP 800-128, Guide for Security-Focused Configuration Management of Information Systems*

- *SP 800-137, Information Security Continuous Monitoring for Federal Information Systems and Organizations*
- *SP 800-147, Basic Input/Output System (BIOS) Protection Guidelines*
- *Draft SP 800-155, BIOS Integrity Measurement Guidelines*
- *FIPS 199, Standards for Security Categorization of Federal Information and Information Systems*

The following table lists the security capabilities provided by the trusted geolocation proof of concept:

Capability Category	Capability Number	Capability Name
IC1 – Measurements	IC1.1	Measured Boot of BIOS
	IC1.2	Measured Boot of VMM
	IC1.3	Baseline for BIOS/VMM Measurements (whitelisting)
	IC1.4	Remote Attestation of Boot Measurements
	IC1.5	Security Capability & Config Discovery
IC2 – Tag Verification	IC2.1	Asset Tag Verification
	IC2.2	Geotag Verification
IC3 – Policy Enforcement	IC3.1	Policy-Based Workload Provisioning
	IC3.2	Policy-Based Workload Migration
IC4 – Reporting	IC4.1	Support for Continuous Monitoring
	IC4.2	Support for On-Demand Reports

The following table maps the security capabilities from the previous table to the NIST SP 800-53 controls in the list at the beginning of this appendix.

	IC1.1	IC1.2	IC1.3	IC1.4	IC1.5	IC2.1	IC2.2	IC3.1	IC3.2	IC4.1	IC4.2
AU-2										X	X
CA-1					X					X	X
CA-2					X					X	X
CA-7										X	X
CM-2			X		X	X					
CM-3	X	X		X		X					
CM-8					X	X					
PE-18							X				
SC-1								X	X		
SC-2								X	X		
SC-4								X	X		
SC-7	X	X			X		X	X	X		
SC-11								X	X		
SC-29		X	X	X	X			X	X		
SC-32						X	X	X	X		
SI-3	X	X	X		X					X	X
SI-4			X	X	X					X	X
SI-6	X	X	X	X	X						
SI-7	X	X	X	X							

Appendix G—Cybersecurity Framework Subcategory Mappings

This appendix maps the major security features of the trusted geolocation proof of concept implementation to the following subcategories from the Cybersecurity Framework:[1]

- ID.GV-1: Organizational information security policy is established
- ID.GV-3: Legal and regulatory requirements regarding cybersecurity, including privacy and civil liberties obligations, are understood and managed
- PR.DS-6: Integrity checking mechanisms are used to verify software, firmware, and information integrity
- PR.IP-5: Policy and regulations regarding the physical operating environment for organizational assets are met

[1] *Framework for Improving Critical Infrastructure Cybersecurity*, Version 1.0, NIST, February 12, 2014. http://www.nist.gov/cyberframework/index.cfm

Appendix H—Acronyms and Other Abbreviations

Selected acronyms and abbreviations used in the report are defined below.

AD	Active Directory
AIK	Attestation Identity Key
API	Application Programming Interface
BIOS	Basic Input/Output System
CA	Certificate Authority
CRTM	Core Root of Trust for Measurement
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DRTM	Dynamic Roots of Trust Measurement
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
GB	Gigabyte
GHz	Gigahertz
GKH	Good Known Host
HBA	Host Bus Adapter
HD	Hard Drive
HTCC	HyTrust CloudControl
IaaS	Infrastructure as a Service
Intel TXT	Intel Trusted Execution Technology
Intel VT	Intel Virtualization Technology
I/O	Input/Output
iSCSI	Internet Small Computer System Interface
ISO	International Organization for Standardization
IT	Information Technology
ITL	Information Technology Laboratory
KVM	Kernel-Based Virtual Machine
MAC	Media Access Control
MLE	Measured Launch Environment
MOB	Managed Object Browser
NC	Nonce
NFS	Network File System
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
OMB	Office of Management and Budget
OS	Operating System
PCR	Platform Configuration Register
PIP	Public IP Address
PXE	Pre-Boot Execution Environment
RAM	Random Access Memory
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SAS	Serial Attached SCSI
SCP	Secure Copy
SFTP	Secure File Transfer Protocol
SML	Stored Measurement Log

SP	Special Publication
SRK	Storage Root Key
TAS	Trust Attestation Service
TFTP	Trivial File Transfer Protocol
TPM	Trusted Platform Module
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMM	Virtual Machine Monitor
VST	Virtual Switch Tagging

Appendix I—References

References for this publication are listed below.

- Evolution of Integrity Checking with Intel® Trusted Execution Technology: an Intel IT Perspective: http://www.intel.com/content/www/us/en/pc-security/intel-it-security-trusted-execution-technology-paper.html

- HyTrust CloudControl Administration Guide: http://downloads.hytrust.com/product_documentation/4.1.0/HyTrust_CloudControl_Administration_Guide.pdf

- HyTrust CloudControl Installation Guide: http://downloads.hytrust.com/product_documentation/4.1.0/HyTrust_CloudControl_Installation_Guide.pdf

- Intel Planning Guide: Cloud Security http://www.intel.com/content/www/us/en/cloud-computing/cloud-security-checklist-planning-guide.html?wapkw=cloud+security+planning+guide

- Intel TXT white paper: http://www.intel.com/content/www/us/en/architecture-and-technology/trusted-execution-technology/trusted-execution-technology-security-paper.html

- OpenStack Icehouse documentation: http://docs.openstack.org/icehouse/