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

NIST Interagency Report 7904
(Draft)

Trusted Geolocation in the Cloud: Proof of Concept Implementation (Draft)

Erin K. Banks
Michael Bartock
Kevin Fiftal
David Lemon
Karen Scarfone
Uttam Shetty
Murugiah Souppaya
Tarik Williams
Raghuram Yeluri

NIST Interagency Report 7904 (Draft)

Trusted Geolocation in the Cloud: Proof of Concept Implementation (Draft)

Erin K. Banks
Michael Bartock
Kevin Fiftal
David Lemon
Karen Scarfone
Uttam Shetty
Murugiah Souppaya
Tarik Williams
Raghuram Yeluri

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

December 2012

U.S. Department of Commerce

Rebecca M. Blank, Acting Secretary

National Institute of Standards and Technology

Patrick D. Gallagher,
Under Secretary for Standards and Technology
and Director

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.

Authority

This publication has been developed by NIST to further its statutory responsibilities under the Federal Information Security Management Act (FISMA), Public Law (P.L.) 107-347. NIST is responsible for developing information security standards and guidelines, including minimum requirements for Federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate Federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), *Securing Agency Information Systems*, as analyzed in Circular A-130, Appendix IV: *Analysis of Key Sections*. Supplemental information is provided in Circular A-130, Appendix III, *Security of Federal Automated Information Resources*.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Interagency Report 7904 (Draft)
42 pages (Dec. 2012)

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. All NIST publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Public comment period: December 21 through January 31, 2013

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930), Gaithersburg, MD 20899-8930
Email: IR7904-comments@nist.gov

Acknowledgments

The authors wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content.

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); virtualization

Trademark Information

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.

1.2 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.

1.3 Document Structure

This document is organized into the following sections and appendices:

- Section 2 defines the problem (use case) to be solved.
- Sections 3, 4, and 5 describe the three stages of the proof of concept implementation.
- Appendix A provides supplementary information from Intel on their portions of the proof of concept implementation.
- Appendix B provides supplementary information from RSA Archer on their portions of the proof of concept implementation.
- Appendix C describes the demonstration architecture including various devices and logical networks.
- Appendix D lists and defines acronyms and other abbreviations used in the publication.

2. Use Case

This section defines the problem—the *use case*—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 use case, 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 very 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.

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. 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, and therefore traditional geolocation methods cannot be trusted to meet cloud security needs.

The motivation behind this use case 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 tamperproof hardware. This information is accessed using secure protocols to assert the integrity of the platform and confirm the location of the host.

2.2 Requirements

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 applications and data. 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 cloud so that the right applications get deployed there.
3. Enable audits of that portion of the cloud so that users can verify compliance.

Such 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

An important component of a solution is having some assurance that the platform the workload is running on can be trusted. If the platform isn't 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., 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 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

- 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, giving it 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., 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

- 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 BIOS (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 spanning different physical locations (e.g., countries 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

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

3. Use Case Instantiation Example: Stage 0

This section describes stage 0 of the proof of concept implementation (platform attestation and safer Virtual Machine Monitor (VMM) launch).

3.1 Solution Overview

This stage of the use case enables the creation of what are called trusted compute pools. Trusted compute pools, also known as trusted pools, 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 resources, or migrated to other trusted platforms within these pools.

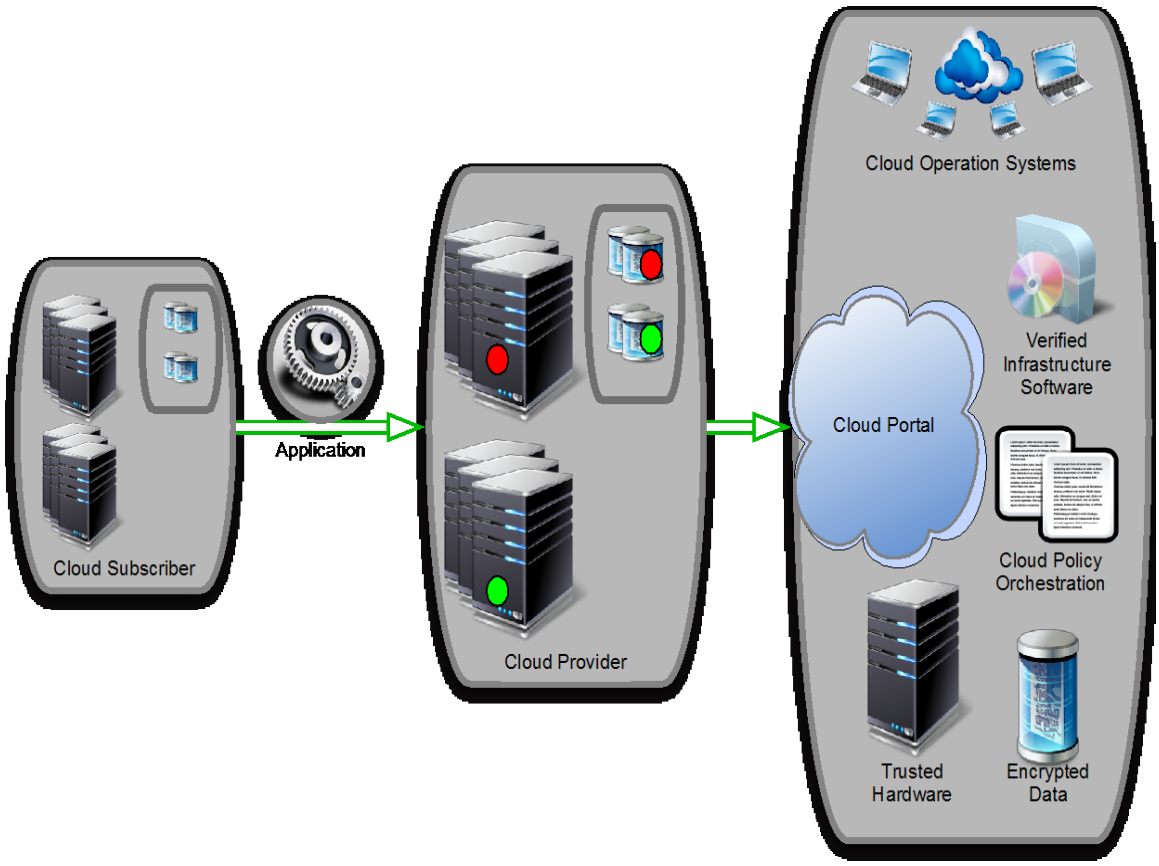


Figure 1: Concept of Trusted Pools

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 Intel TXT and if a specific platform has a defined/compliant OS/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 service on a set of available nodes/servers, believe the response from the platform?

Attestation provides the definitive answers to these questions. 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 Intel TXT provides 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 Trusted Platform Module (TPM) provides these.

Attestation is the process of providing a digital signature of a set of platform configuration registers (PCRs)—a set of registers in a TPM that are extended with specific measurements for various launch modules of the software—and having the requestor validate the signature and the PCR contents. The entity wishing to validate requests the TPM_Quote command specifying an Attestation Identity key to perform the digital signature, the set of PCRs to quote, and a nonce to ensure freshness of the digital signature. Figure 2 illustrates the attestation process and flow.

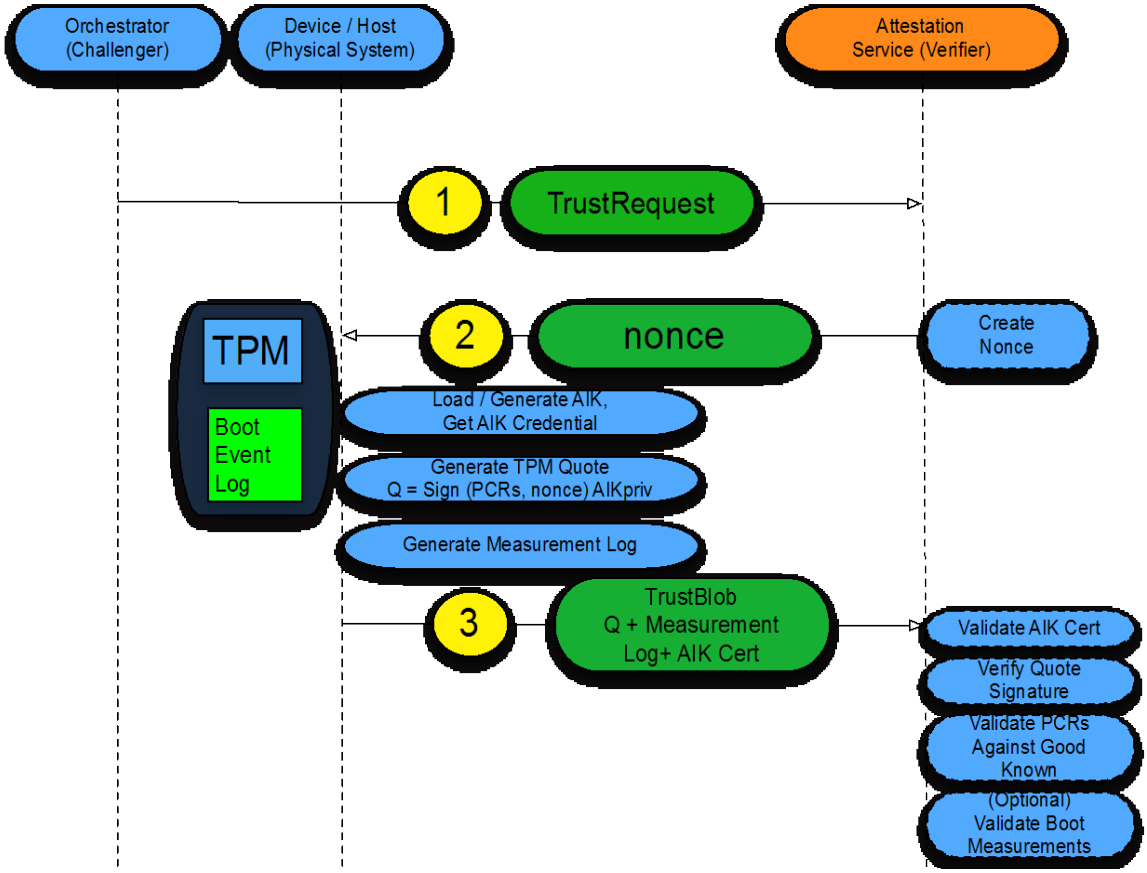


Figure 2: Attestation Process and Flow

The entity that challenged the platform for this information now can make a determination about the trust of the launched platform by comparing the measurements from the TPM quote 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 a

service delivered as a service by a trusted third party for service providers and enterprises to use.

3.2 Intel Trusted Execution Technology (Intel TXT)

Intel Trusted Execution Technology (Intel TXT) provides 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:

- Trusted extensions integrated into silicon (processor and chipset). These instructions allow for the orderly quiescence of all activities on the platform such that a tamper-resistant environment is enabled for the measurement and verification process and allows for protection of platform secrets in the case of "reset" and other disruptive attacks.

- Authenticated code modules (ACMs). Platform-specific code is authenticated to the chipset and executed in an isolated environment within the processor and the trusted environment (authenticated code mode) enabled by ACMs to perform secure tasks.

- Launch control policy (LCP) tools. LCP provides the local definition of the "list" of "known good" configurations and components—this provides the foundational definition that the platform vendor or owner will consider their trusted platform.

Some of the required components for the Intel TXT secured platform are provided by third parties, including:

- Trusted Platform Module (TPM) 1.2 (third party silicon). The TPM is a hardware device defined by the Trusted Computing Group that stores authentication credentials in platform configuration registers (PCRs), which are issued by Intel TXT.

- Intel TXT-enabled BIOS, firmware, operating system, and hypervisor environments.

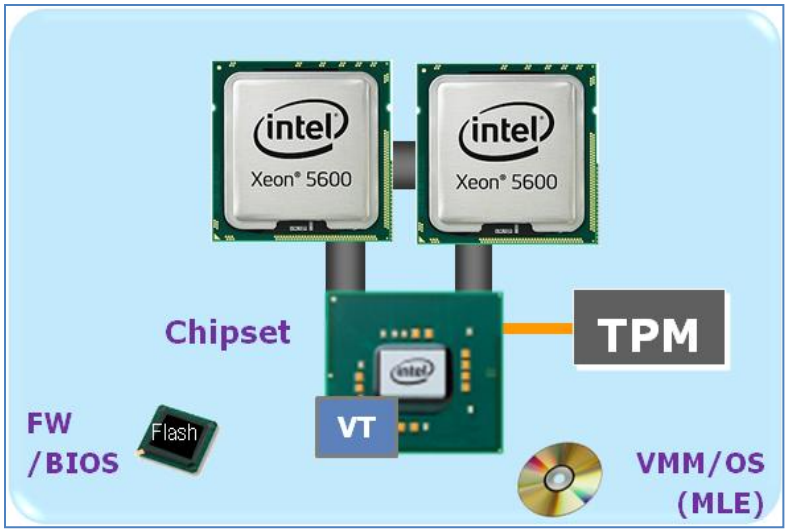


Figure 3: Intel TXT Components

The capabilities of Intel TXT include:

- 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.

3.3 Solution Architecture

Figure 4 provides a layered view of the solution system architecture, beginning with TXT enabled servers based on the Intel Xeon 5600 family of processors. Intel TXT-enabled servers also include a hardware security module (TPM) for storing sensitive keys and measurements. VMware ESXi was used as the OS/hypervisor. All the servers were configured with VMware vCenter 4.1 U1 server.

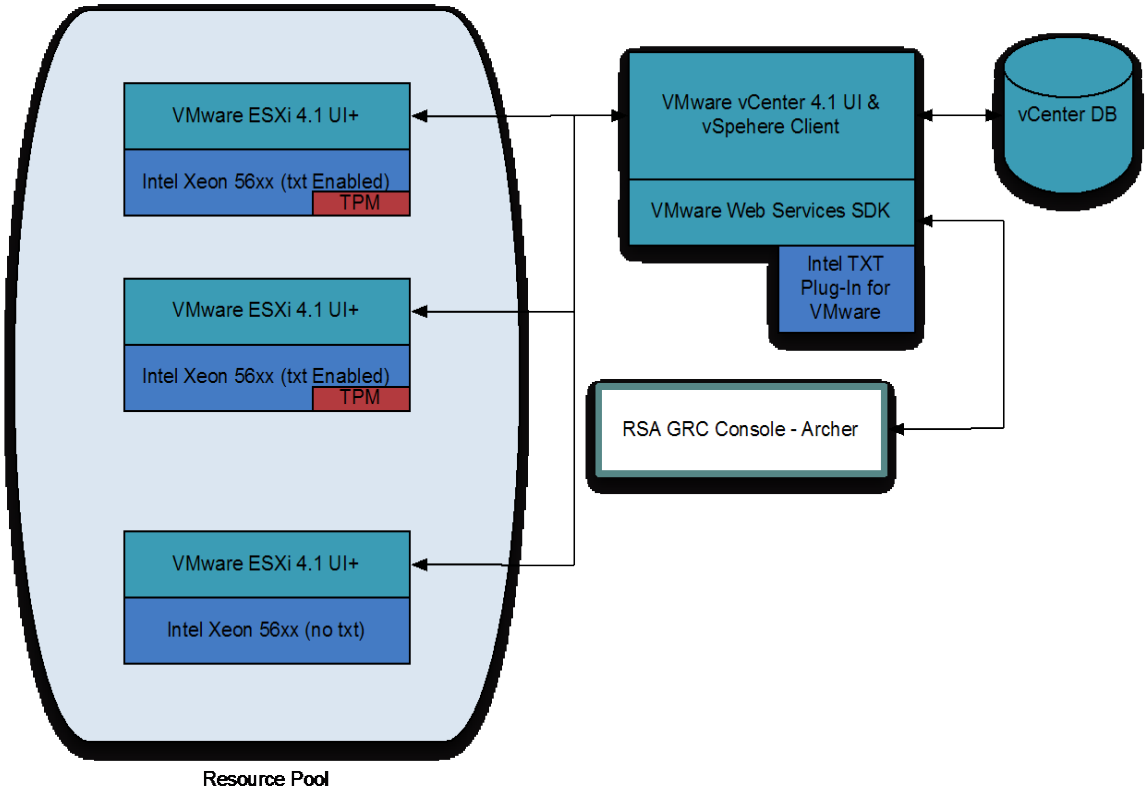


Figure 4: Stage 0 Solution System Architecture

The initial step in instantiating the architecture requires provisioning the server for TXT. This currently needs physical access to the server to access the BIOS, enable a set of configuration options to use the TPM, and activate TXT. This process is highly BIOS and OEM dependent. This step is mandatory for a measured launch of the OS/hypervisor.

VMware has implemented support for Intel TXT starting with VMware ESXi 4.1. With this support, if TXT has been enabled in hardware, the ESXi hypervisor undergoes a measured launch, and the BIOS and VMM components are measured (cryptographically) and extended into the server TPM's PCRs. These measurement values stored in the TPM are accessible through VMware vCenter via the VMware Web Services SDK. When the hosts are initially configured with vCenter server using the vSphere Client, the values stored in the PCRs are cached in the vCenter database.

In order to access the measurements (and provide basic attestation) for a given ESXi Server from within vCenter for decision making, Intel has provided a reference plug-in to vCenter called the Intel TXT plug-in. This is registered with vCenter server using the VMware ExtensionManager APIs. This plug-in is a web application as per the requirement of the vCenter for extending its functionality. The plug-in retrieves the PCR information through vSphere Web Services SDK, compares it with the good known values, and shows the corresponding status in the plug-in UI.

In addition to the measured launch, this solution architecture also provides provisions to assign a secure geolocation tag to each of the servers during the provisioning process (note: this is a feasibility prototype feature, and not a commercially available feature yet). The tag is provisioned to a non-volatile index in

the TPM via an out-of-band mechanism, and on a TXT-enabled hypervisor launch, the contents of the index are inserted/extended into one of the PCRs in the TPM. Figure 5 illustrates the geotag provisioning and usage flow. Intel TXT plug-in (described in the previous paragraph) also provides the interface and attestation to the geotag information, including the geotag lookup and user-readable/presentable string/description.

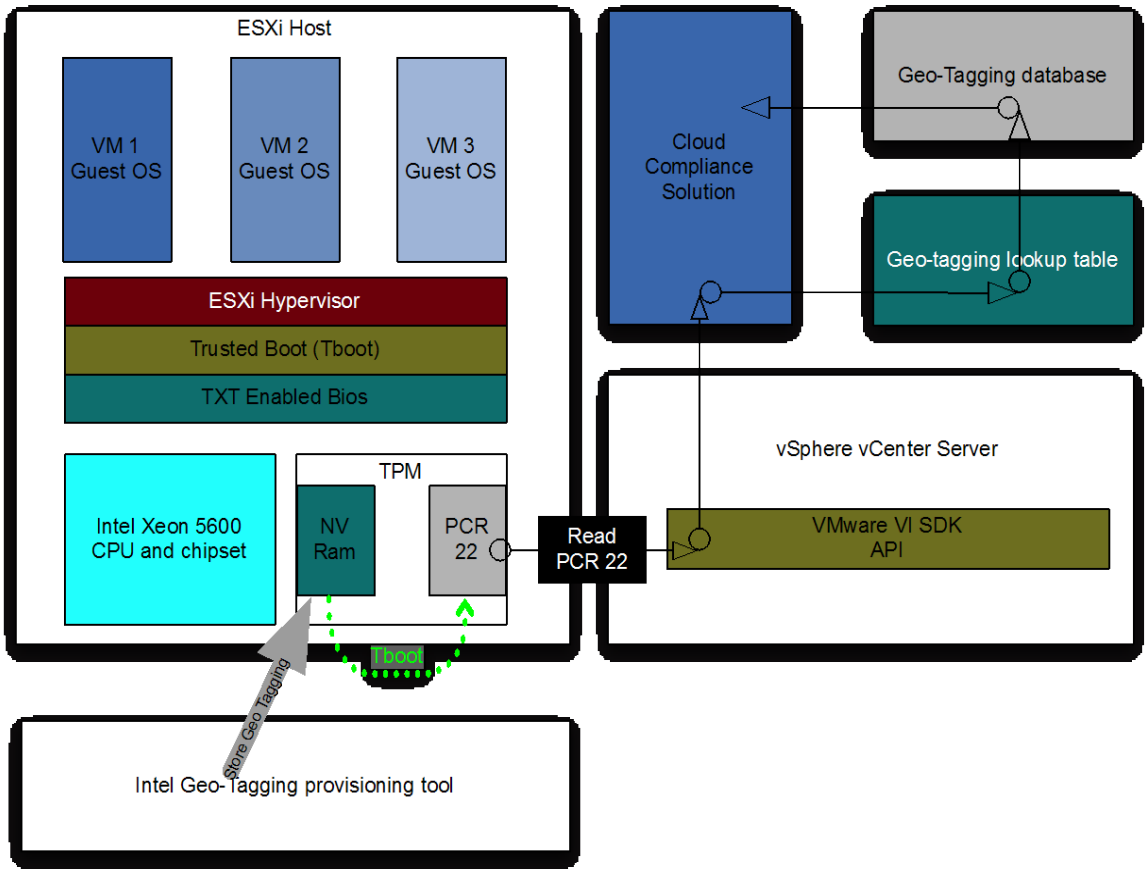


Figure 5: Geotag Provisioning and Usage Flow

4. Use Case Instantiation Example: Stage 1

This section discusses stage 1 of the proof of concept implementation (trust-based and geolocation-based homogeneous secure migration), which is based on the stage 0 work and adds components that monitor and enforce geolocation-based restrictions.

4.1 Solution Overview

Figure 6 shows the operation of the stage 1 solution. It assumes that Server A and Server B are two servers within the same cloud.

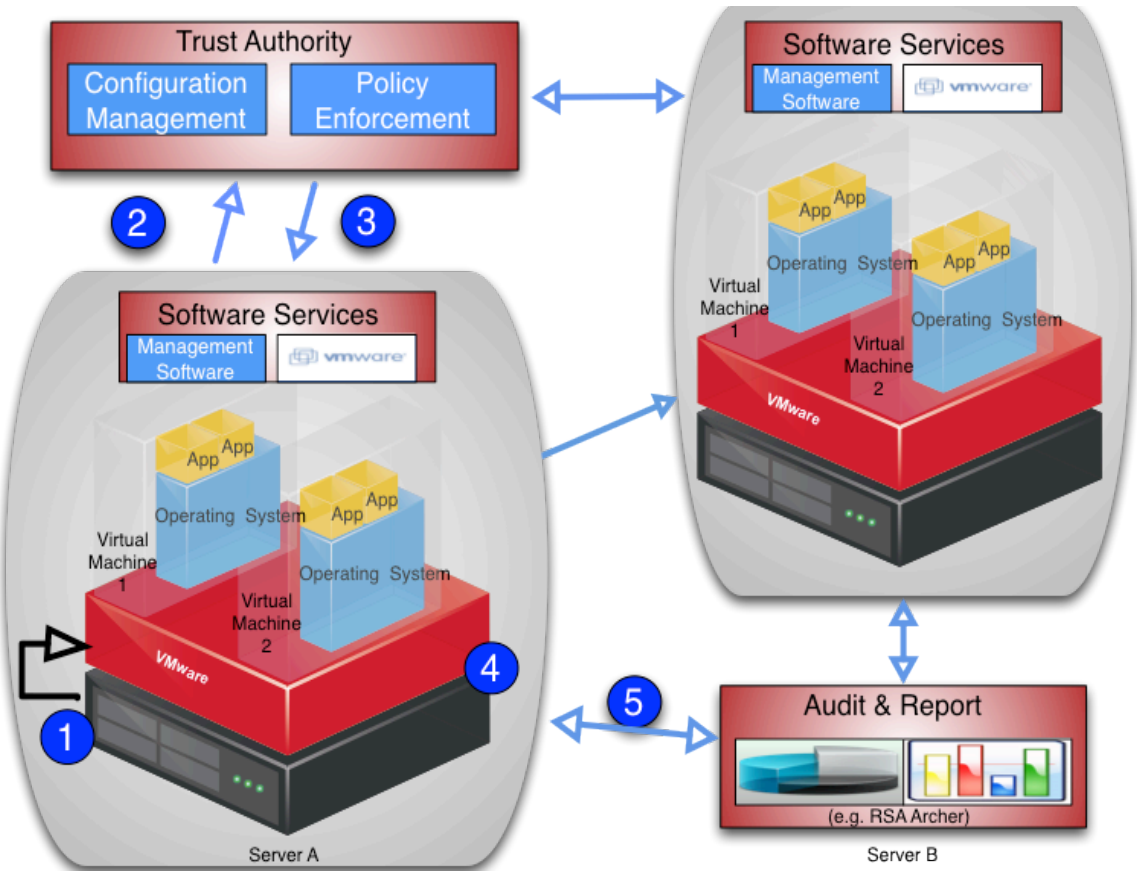


Figure 6: Stage 1 Solution Overview

There are five generic steps performed in the operation of the stage 1 solution, as outlined below and reflected by the numbers in Figure 6:

1. Server A performs an Intel TXT measured launch, with Intel TXT populating the PCR values.

2. Server A sends a TPM quote to the Trust Authority. The TPM quote includes signed hashes of the BIOS, TBOOT, ESX, 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 PCR values.

4.2 Solution Architecture

Figure 7 shows the stage 1 architecture. Note that this is identical to the stage 0 architecture, with additional measurement occurring related to the geolocation policy.

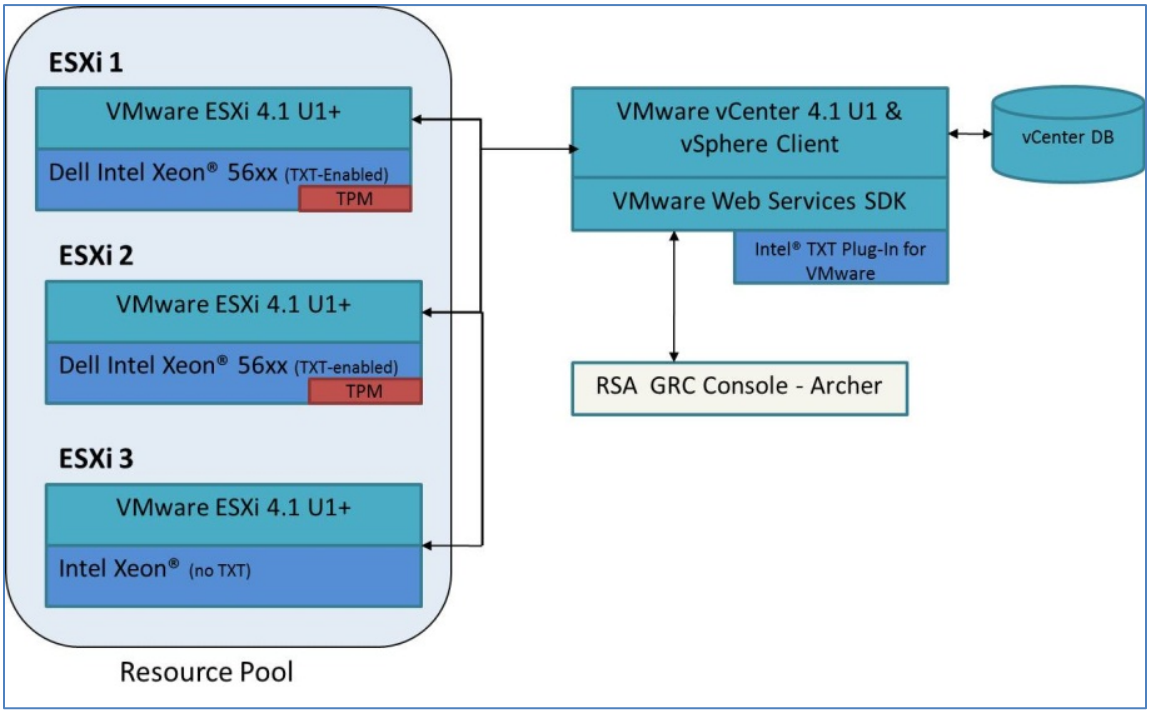


Figure 7: Stage 1 Architecture

5. Use Case Instantiation Example: Stage 2

This section discusses stage 2 of the proof of concept implementation (trust-based and geolocation-based homogeneous secure migration driving a governance, risk, and compliance dashboard), which is based on the stage 1 work and adds components that gather and display the measurements in a governance, risk, and compliance dashboard.

5.1 Solution Overview

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

Figure 8 shows a chart reflecting the relative size of the pools of trusted (green) and untrusted (red) cloud servers. In this example, there are two servers in the trusted pool and one server in the untrusted pool.

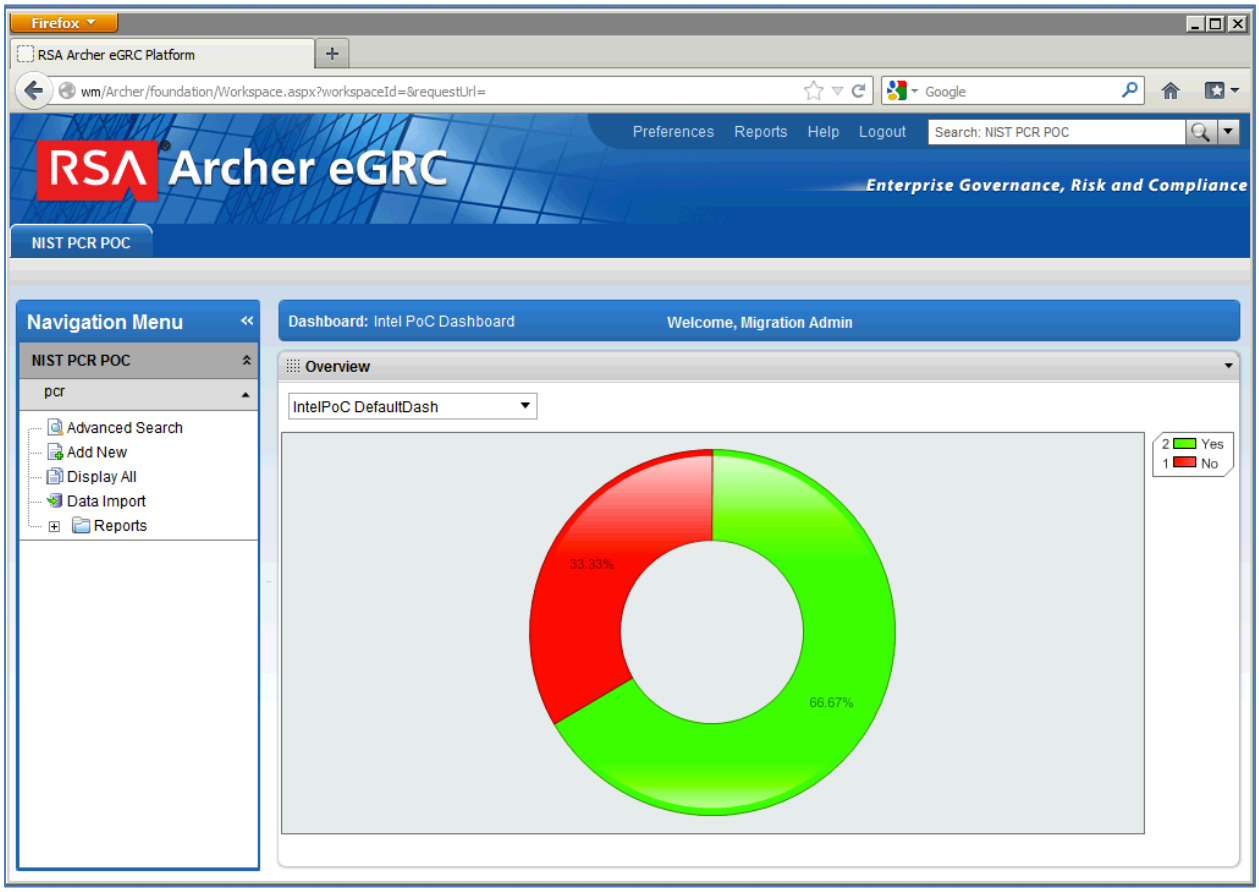


Figure 8: Default Dashboard View

Figure 9 is a drill-down page from the high-level dashboard view shown in Figure 8. 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. Based on the colors of the measurement statuses, with green representing a positive status and red representing a negative status, the first two servers are trusted for use because they pass all the status checks, and the third server is untrusted because it does not pass all the status checks. Note that the system validation measurement is a summary measurement that takes into consideration the status of the other measurements. The system validation measurement will be green if all the other measurements are green, otherwise red.

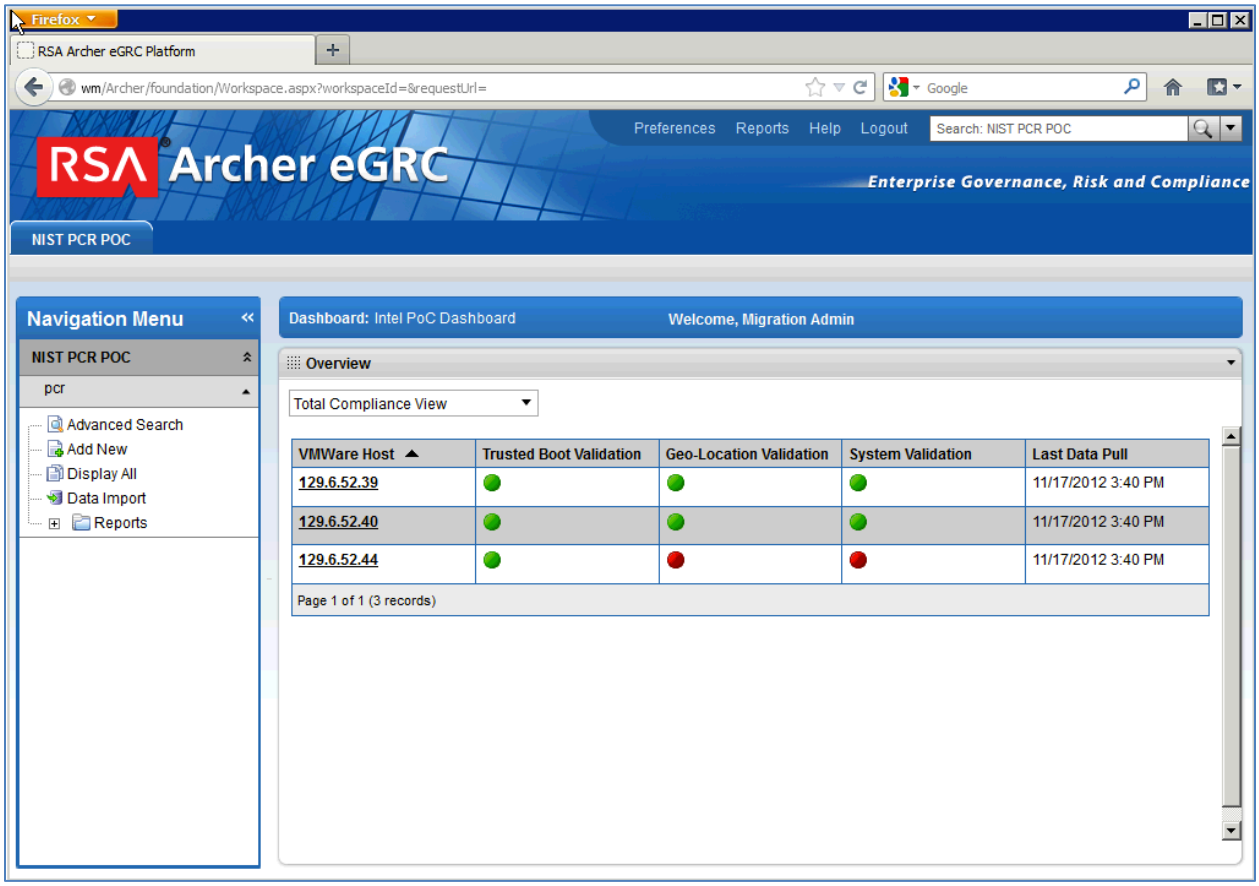


Figure 9: Trusted Boot and Geolocation Compliance View

Figure 10 shows a drill-down from Figure 9 for an individual server, in this example the first server listed in Figure 9. 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.

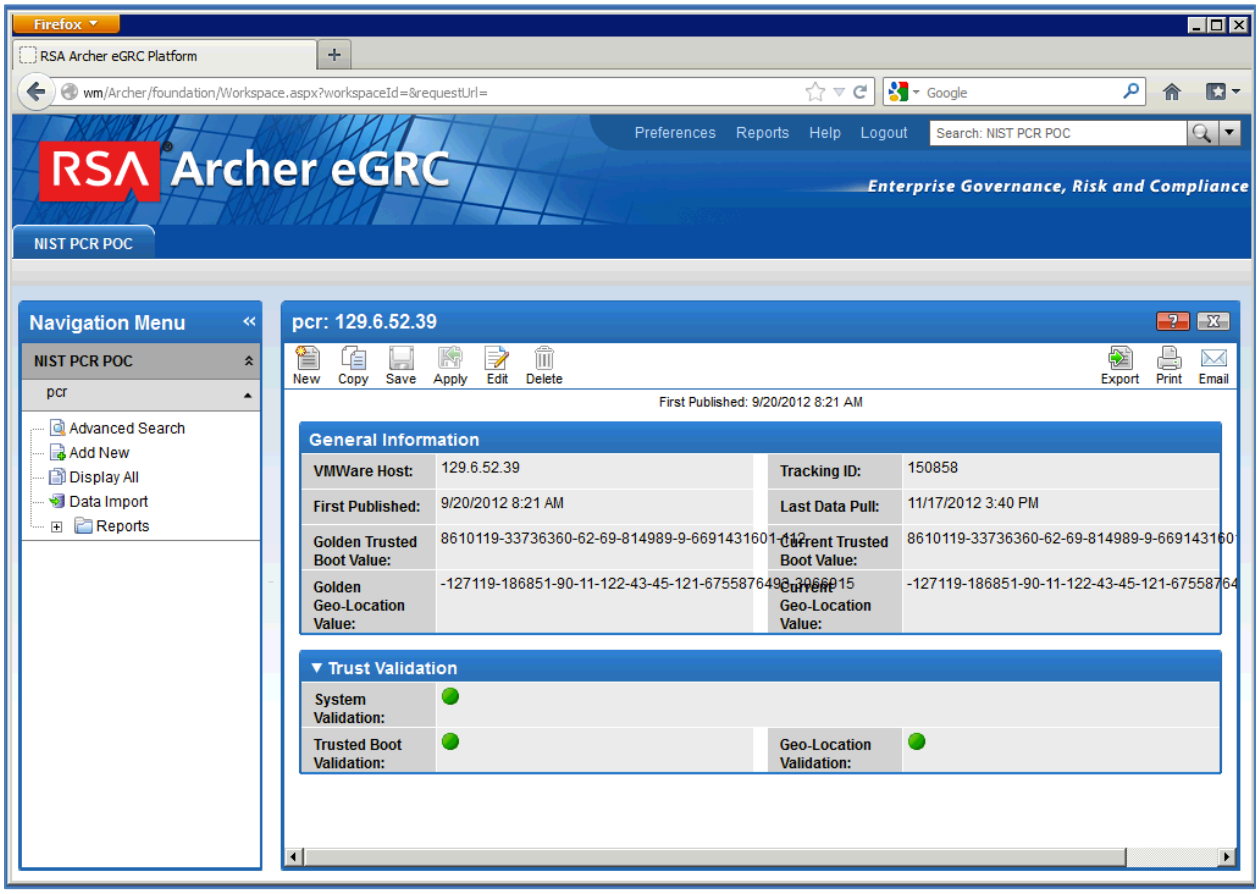


Figure 10: Single Server Overview

5.3 Solution Architecture

Figure 11 shows the stage 2 architecture. Note that this is identical to the stage 0 and stage 1 architectures, with additional reporting and monitoring occurring related to governance, risk, and compliance.

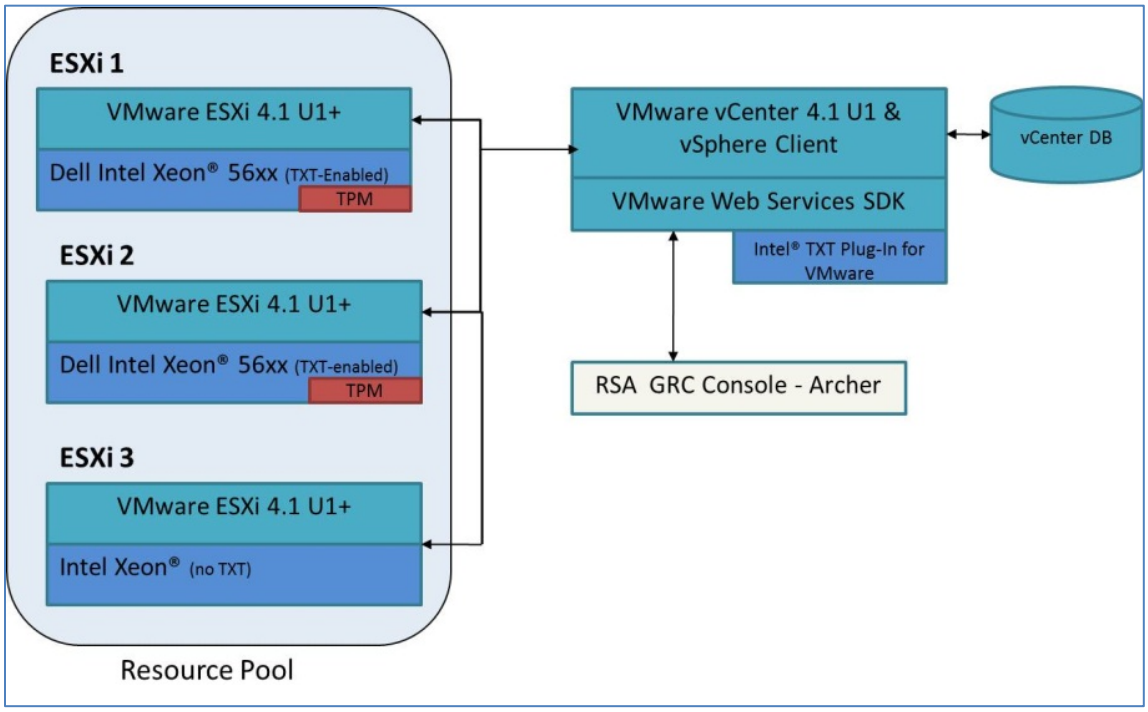


Figure 11: Stage 2 Architecture

Appendix A—Intel Supplemental Information

This section contains supplementary information provided by Intel describing all the required components and steps required to setup the POC.

A.1 Hardware Description

Table 1: Hardware Description

System	Processor Configuration	Detailed Configuration
1 Management Server • Microsoft Windows8 2008, IIS, .NET 2.0 • VMware vCenter Server 4.1 • VMware vSphere Web Services SDK	Intel Xeon Processor L5630 See processor details at http://ark.intel.com/Product.aspx?id=47927	Form Factor: 2U Rack Mount Server Processor: Intel Xeon Processor X5600 @2.93 GHz, 2-socket x 6 cores = 12 cores Memory: 24 GB RAM Storage: 100 GB HDD 10Gb Ethernet network
1 Management Client • Microsoft Windows 7 • VMware vSphere Client 4.1	Intel Xeon Processor L5630 See processor details at http://ark.intel.com/Product.aspx?id=47927	Form Factor: 2U Rack Mount Server Processor: Intel Xeon Processor X5600 @2.93 GHz, 2-socket x 6 cores = 12 cores Memory: 24 GB RAM Storage: 100 GB HDD 10Gb Ethernet network
3 ESXi Host (x2) ☐ VMware ESXi	Intel Xeon Processor L5630 See processor details at http://ark.intel.com/Product.aspx?id=47927	Form Factor: 2U Rack Mount Server Processor: Intel Xeon Processor X5600 @2.93 GHz, 2-socket x 6 cores = 12 cores Memory: 24 GB RAM Storage: 100 GB HDD 10Gb Ethernet network
1 iSCSI Data Store Server	Dell PowerVault 3200i	iSCSI device

A.2 BIOS Changes

The following changes are required in the BIOS settings:

1. Intel TXT set to “Enabled”.
2. Intel Virtualization Technology (Intel VT) set to “Enabled”.
3. Intel VT for Directed I/O set 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.

A.3 VMware Components

The high-level installation and configuration steps for the infrastructure setup required to employ the Intel TXT capabilities supported by the platform are listed below. These setup steps assume a basic understanding of how to install and configure Windows Server 2008 R2 Enterprise, VMware vCenter Server, and VMware vSphere Client.

1. Install Windows Server 2008 R2 Enterprise on compatible hardware.

2. Install the VMware vCenter Server.

3. Set up the vSphere client machine:

    a. Install Windows 7 on compatible hardware.

    b. Launch a browser and point it at the machine where vCenter Server is installed to download the vSphere client.

    c. Install the VMware vSphere client.

4. Install the VMware ESXi hosts:

    a. Install VMware ESXi on the hosts. This version supports Intel TXT

    b. Ensure that the installation of the hypervisor is initiated after required BIOS settings are configured.

5. After the hypervisor installation completes, add the host to a cluster via the vSphere Client.

6. Once completed, ensure t-boot is enabled. Do this by selecting the host's Configuration tab > "Software" > "Advanced Settings" > "Misc" and set the option "Misc.enableTboot" parameter to "1". See Figure 12 for a screen shot of the settings pane.

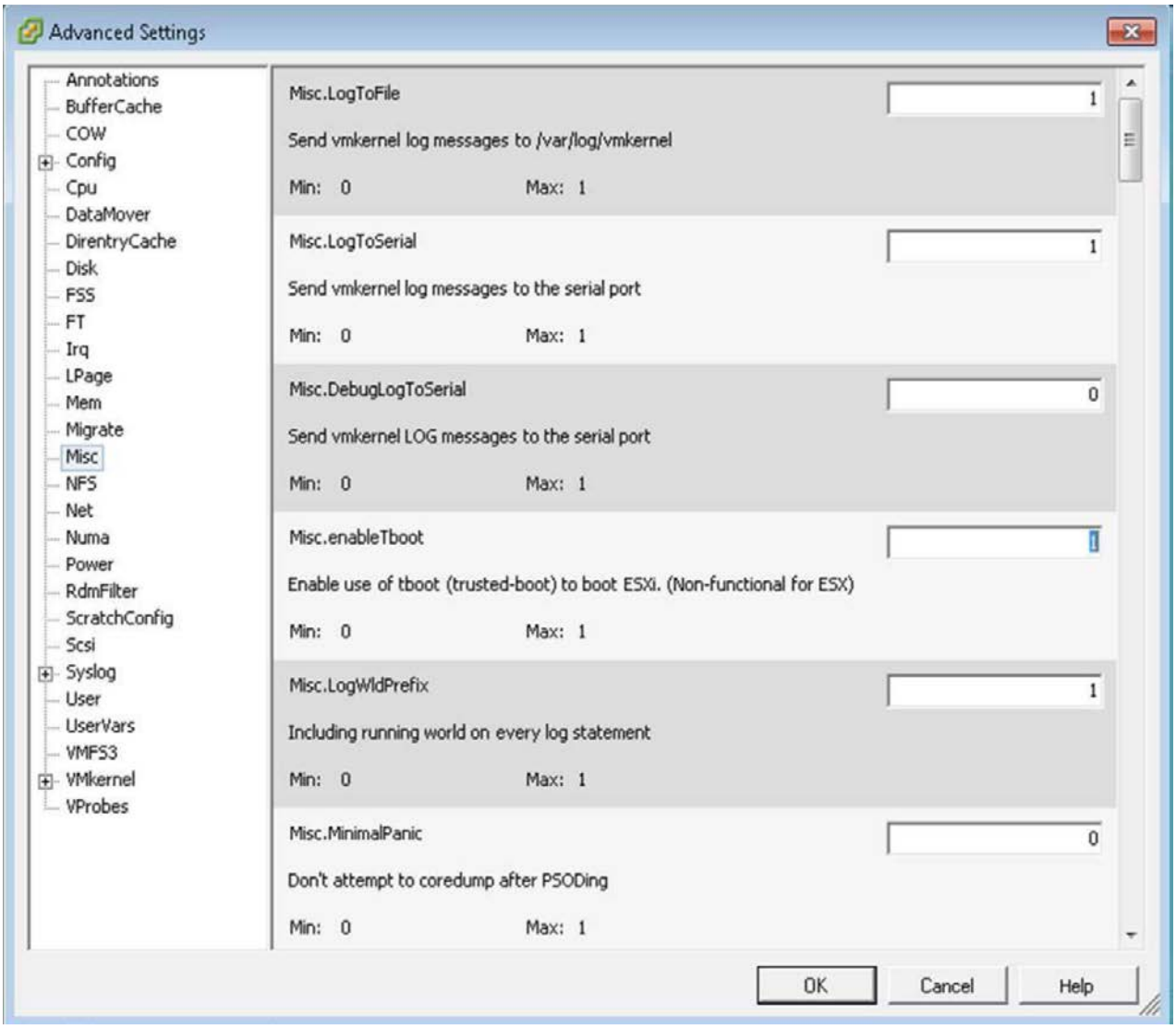


Figure 12: Enable Misc.enableTboot

7. Reboot the host and check that it has booted into trusted mode. Use the Managed Object Browser tool to verify that the "vmwarevmkernel" object's "HostTpmDigestInfo" is present under the "HostRuntimeInfo" of the ESXi host. Figures 13 and 14 show how to verify these values.

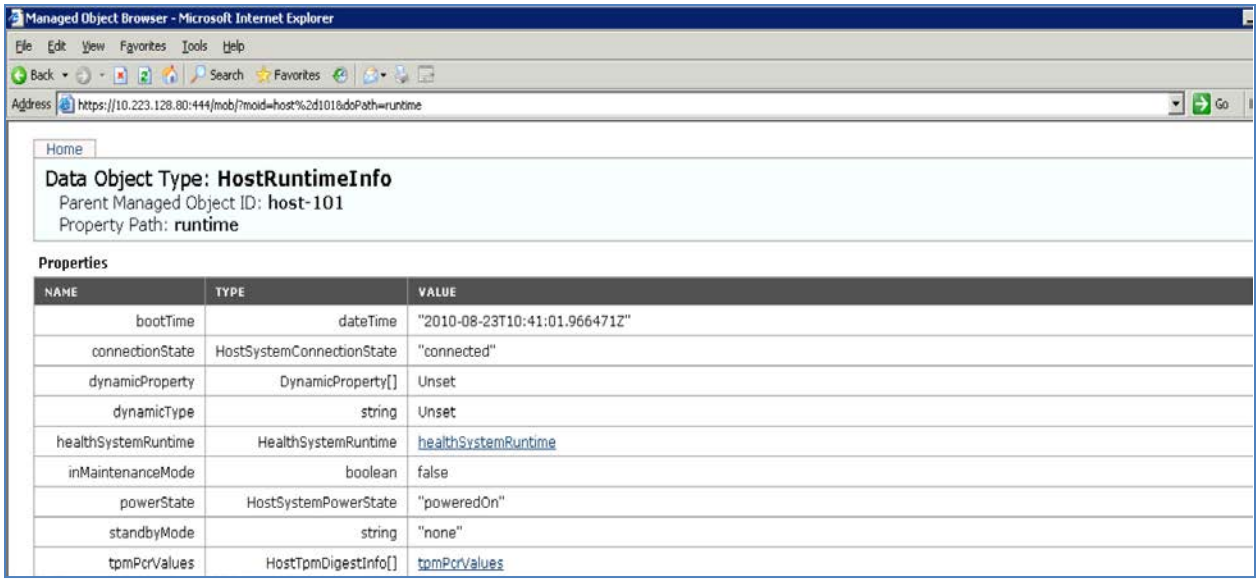


Figure 13: Data Object Type: HostRuntimeInfo

HostTpmDigestInfo

NAME	TYPE	VALUE
digestMethod	string	"SHA1"
digestValue	byte[]	• 47 • 64 • 4 • -2 • -106 • -128 • -61 • -88 • 96 • 57 • -96 • -51 • -98 • 99 • -65 • 37 • 25 • 30 • 109 • 59
dynamicProperty	DynamicProperty[]	Unset
dynamicType	string	Unset
objectName	string	"vmware-vmkernel"
pcrNumber	int	8

Figure 14: HostTpmDigestInfo: Verify Digest Value

8. Configure VMware vCenter Server. Create a single cluster and add all VMware ESXi hosts.

A.4 Plug-In Installation and Configuration

Now that the VMware components have been installed, the Intel TXT demo plug-in needs to be installed next.

1. Launch the IntelTXTDemoPlugIn setup. If needed, change the name of the "Virtual Directory", as shown in Figure 15. Complete the installation.

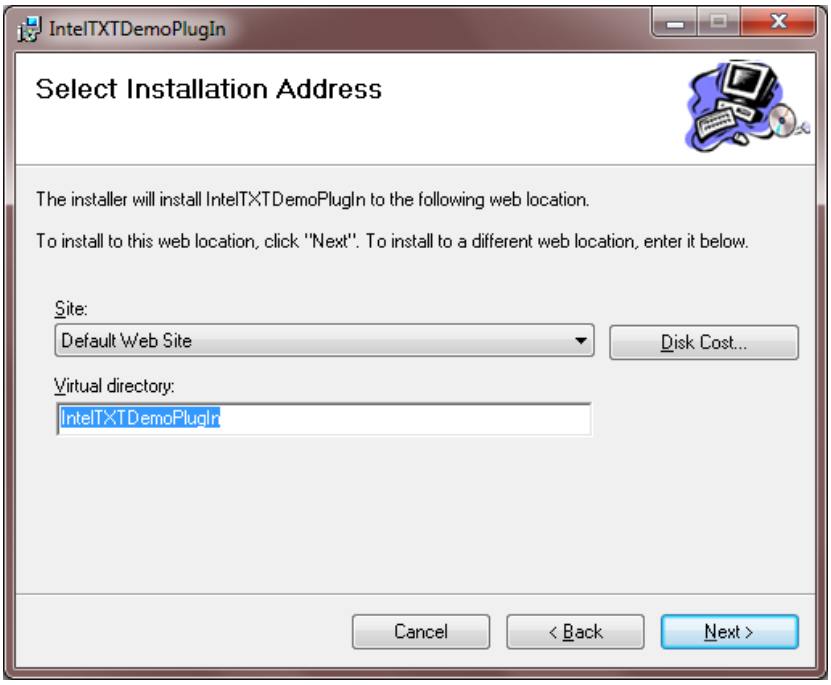


Figure 15: IntelTXTDemoPlugIn Virtual Directory

2. Now we need to edit the web.config file located under the installation website. If the virtual directory was changed to "IntelTXTDemoPlugIn", then the web.config file would be located under the "C:\inetup\wwwroot\IntelTXTDemoPlugIn" folder. The appsettings section of the web.config file needs to be modified to reflect the environment on which the plug-in has been installed.

```
<add key="VCSDK_URL" value="https://127.0.0.1:444/sdk" />
<add key="UserName" value="Administrator" />
<add key="Password" value="P@ssw0rd" />
<add key="LocationCompliance" value="1"/> <!-- Set this to 1 to enable policy engine or to 0 --
<add key="AuditLogging" value="0"/> <!-- Set this to 1 to enable policy engine or to 0 -->
<add key="ImagesURL" value="http://127.0.0.1/IntelTXTDemoPlugIn/Images/" />
<add key="ESXHostConfigFile" value="C:\Inetpub\wwwroot\IntelTXTDemoPlugIn\ESXHosts.xml" />
<add key="GeoLocationFile" value="C:\Inetpub\wwwroot\IntelTXTDemoPlugIn\GeoLocation.xml" />
<add key="AuditLogFile" value="C:\tmp\AuditLog.txt" />
```

Below are the changes that need to be done for the web.config file:

a. If you have installed the plug-in on the same server as vCenter server, the "VCSDK_URL" need not be changed because it is pointing to the local server. Update the port number if needed.

b. Update the "UserName" & "Password" values for your environment.

c. If you have provisioned the ESXi host for geotagging, set the "LocationCompliance" to "1", else set to "0".

d. AuditLogging is a feature that would write to a file the information and error messages thrown by the plug-in. Enable it if needed. If you enable it, ensure that the "AuditLogFile" is pointing to the right file.

e. Update the "ImagesURL" to reflect the virtual directory in which the plug-in was installed.

f. Ensure that both "ESXHostConfigFile" and "GeoLocationFile" fields are pointing to the right location based on the installation folder.

3. Edit the ESXHosts.XML file and ensure that there are entries for each of the TXT based systems that are configured in the environment. This file will be used to store the good known values of the ESXi hosts. Clear the values of "digest" and "bootTime" fields if it is already not empty. The plug-in will populate the same when it is loaded for the first time. NOTE: Please do NOT delete the quotes. Just delete the contents within the quotes as shown below.

```
<?xml version="1.0" encoding="utf-8" ?>
- <ESXHosts>
    <ESXHost name="" Location="" digest="" connectionState="poweredOn" bootTime="" />
    <ESXHost name="" Location="" digest="" connectionState="poweredOn" bootTime="" />
  </ESXHosts>
```

4. Modify the MOB_ExtMgr.XML file to point the URL field to the right installation folder so that it is pointing to the right XML file, which actually has the URL of the plug-in. The contents of this file would be used to register the plug-in with VMware vCenter.

```
<extension>
  <description>
    <label>com.Intel.TXT.VC.PlugIn</label>
    <summary>Intel TXT Demo PlugIn</summary>
  </description>
  <key>com.Intel.TXT.VC.PlugIn</key>
  <version>1.0.0</version>
  <subjectName>/O=Intel /OU=CDE/CN=VI Client plugin</subjectName>
  <server>
    <url>http://127.0.0.1/IntelTXTDemoPlugIn/TXTPlugIn.xml</url>
    <description>
      <label>com.Intel.TXT.VC.PlugIn</label>
      <summary>Intel TXT Demo PlugIn</summary>
    </description>
    <company>Intel</company>
    <type>com.vmware.vim.viClientScripts</type>
    <adminEmail>sudhir.s.bangalore@intel.com</adminEmail>
  </server>
  <client>
    <version>4.0.0</version>
    <description>
      <label>com.Intel.TXT.VC.PlugIn</label>
      <summary>Intel TXT Demo PlugIn</summary>
    </description>
    <company>Intel</company>
    <type>Script Plugin</type>
    <url></url>
  </client>
  <lastHeartbeatTime>2009-06-18T04:10:22.25Z</lastHeartbeatTime>
</extension>
```

URL to be updated to reflect the installation folder

5. Update the TXTPlugIn.xml file to reflect the correct installation website. The title field can be modified if needed. This is the text that will be used to create the tab in the vSphere client.

```
<scriptConfiguration version="1.0.0">
  <key>com.Intel.TXT.VC.PlugIn</key>
  <description>Intel TXT Demo Plug-In</description>
  <view parent="Inventory.Datacenter">
    <title locale="en">Intel TXT Demo Plug-In</title>
    <url>http://127.0.0.1/IntelTXTDemoPlugIn/Default.aspx</url>
  </view>
</scriptConfiguration>
```

Title of the tab in the vSphere client

URL to be updated to reflect the installation folder

6. Create two policy attributes for virtual machines. These policy attributes would be used by the plug-in for making VM migration decisions. For this, highlight one of the Virtual Machine names, click on the corresponding “Summary” tab, and scroll down and click on the “Edit” option of “Annotations”, as shown in Figure 16.

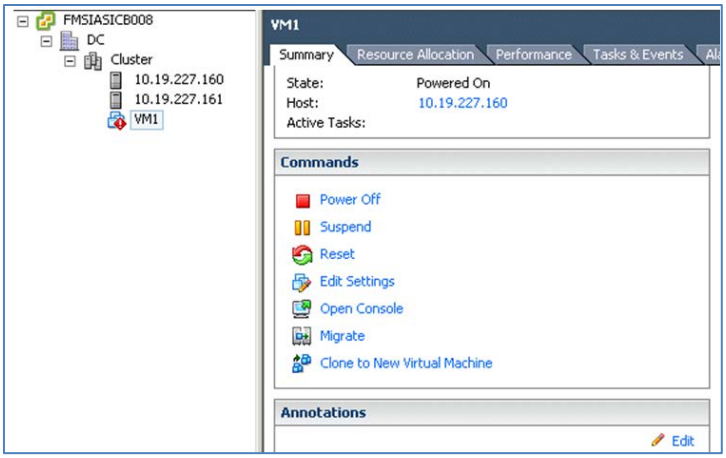


Figure 16: Virtual Machine Summary Tab

Then create the two custom attributes: "TrustedBoot" and "LocationCompliance", and set them to "0", as shown in Figure 17. NOTE: No spaces are allowed in the attributes.

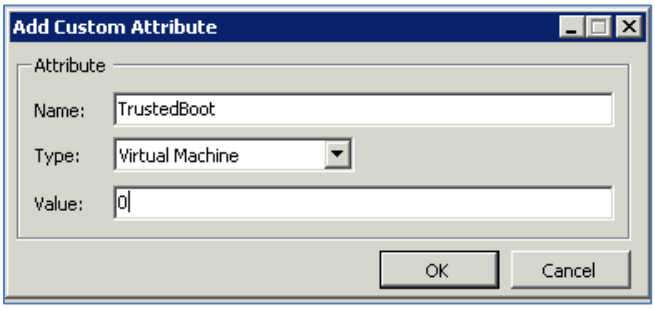


Figure 17: Add Custom Attribute

The final list of custom attributes should look similar to the screenshot in Figure 18.

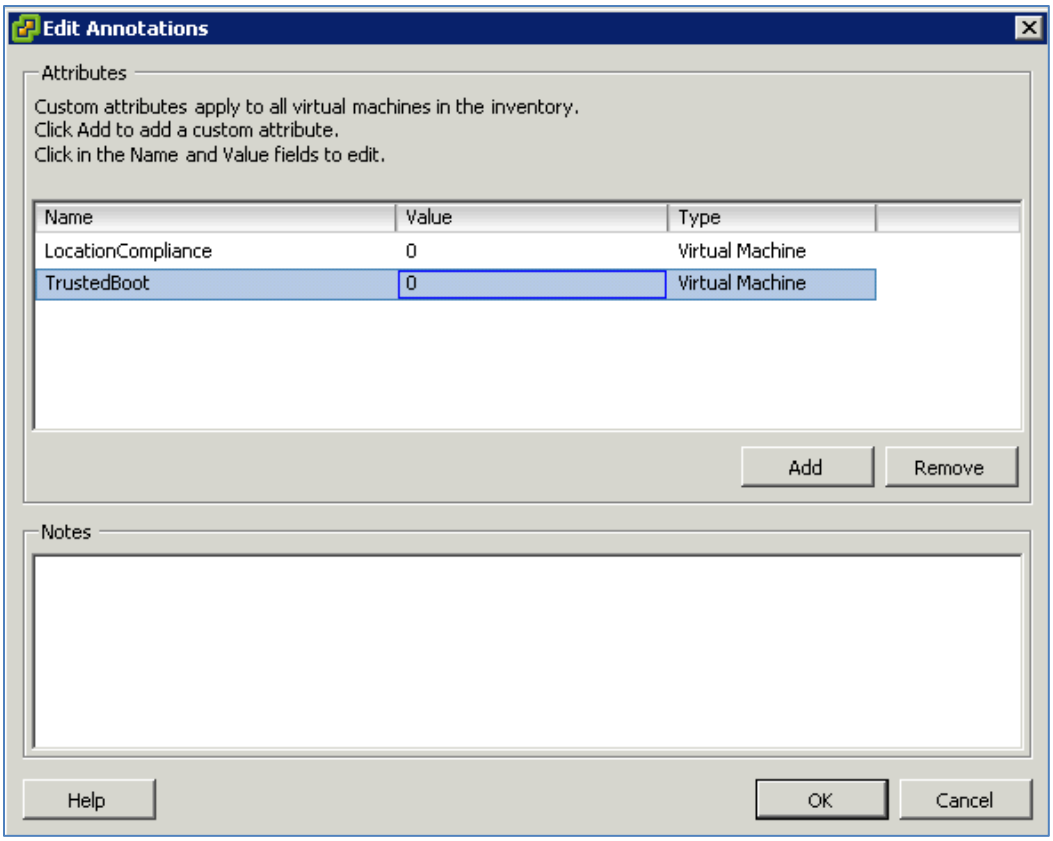


Figure 18: Edit Annotations

A.5 Plug-In Registration

Now that the plug-in has been installed successfully, we need to register this plug-in with the vCenter server.

1. For registering the plug-in, use the tool provided by VMware called "Managed Object Browser (MOB)". This is installed by default on the same server as vCenter. The URL for accessing the MOB is https://127.0.0.1:444/mob. This is using the localhost IP address, which should be replaced with the actual IP address if the vCenter server is being accessed from outside the vCenter server. Note that the port number that is being used is 444, which was configured during installation.

2. When you open the browser (Internet Explorer) and go to the MOB tool URL, you would get the certificate error shown in Figure 19, which needs to be ignored and continued.

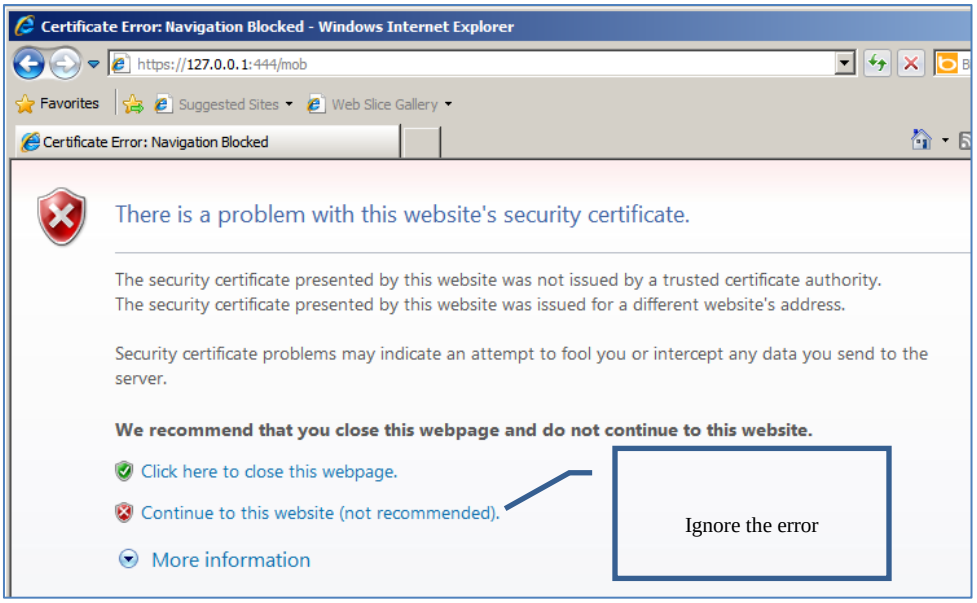


Figure 19: Certificate Error

3. Enter the vCenter administrative credentials, as shown in Figure 20, and continue.

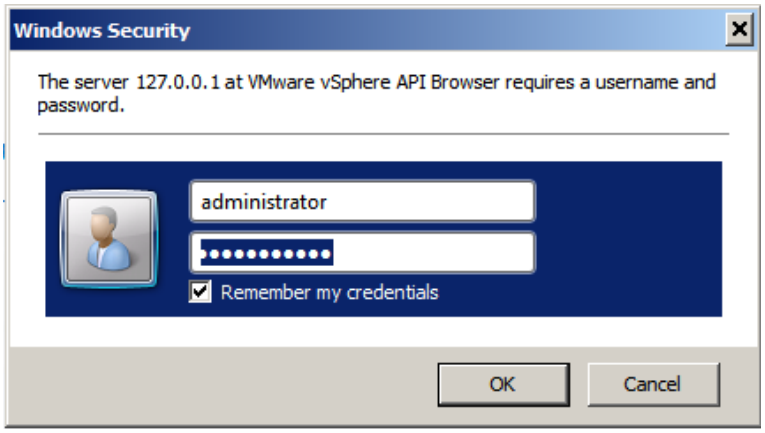


Figure 20: vCenter Administrative Credentials

4. Click on the "Content" link on the home page after successful login. The link can be seen in Figure 21.

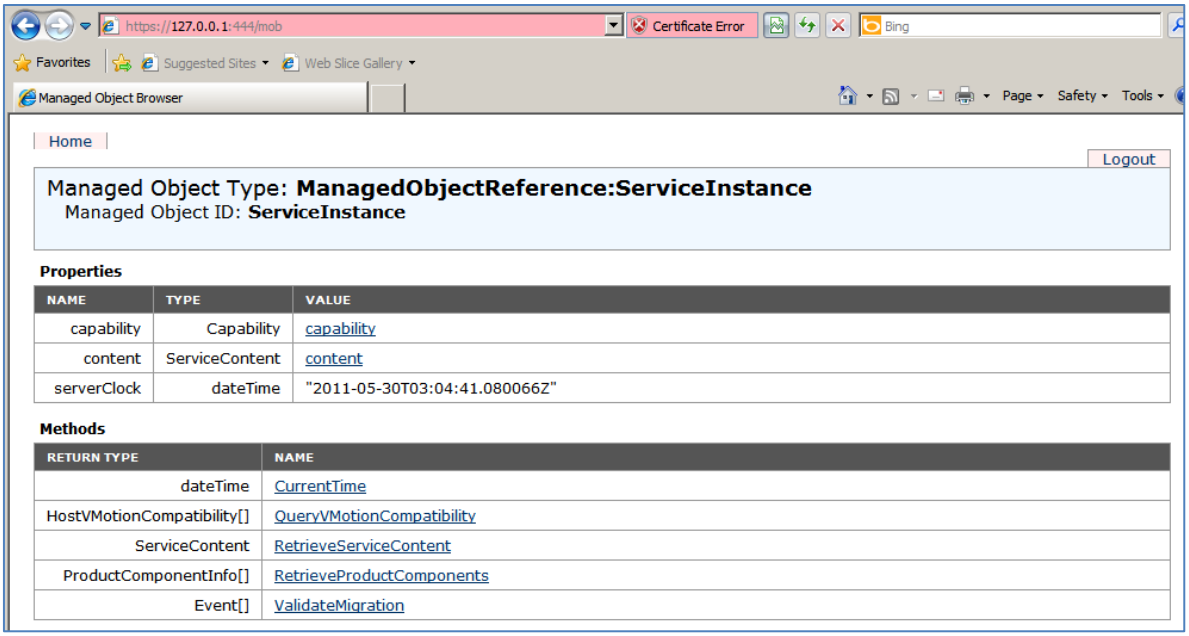


Figure 21: Managed Object Browser

5. Now click on the “Extension Manager” link, as shown in Figure 22.

Home

Data Object Type: ServiceContent
Parent Managed Object ID: **ServiceInstance**
Property Path: **content**

Properties

NAME	TYPE	VALUE
about	AboutInfo	about
accountManager	ManagedObjectReference:HostLocalAccountManager	Unset
alarmManager	ManagedObjectReference:AlarmManager	AlarmManager
authorizationManager	ManagedObjectReference:AuthorizationManager	AuthorizationManager
clusterProfileManager	ManagedObjectReference:ClusterProfileManager	ClusterProfileManager
complianceManager	ManagedObjectReference:ProfileComplianceManager	MoComplianceManager
customFieldsManager	ManagedObjectReference:CustomFieldsManager	CustomFieldsManager
customizationSpecManager	ManagedObjectReference:CustomizationSpecManager	CustomizationSpecManager
diagnosticManager	ManagedObjectReference:DiagnosticManager	DiagMgr
dvSwitchManager	ManagedObjectReference:DistributedVirtualSwitchManager	DVSManager
dynamicProperty	DynamicProperty[]	Unset
dynamicType	string	Unset
eventManager	ManagedObjectReference:EventManager	EventManager
extensionManager	ManagedObjectReference:ExtensionManager	ExtensionManager
fileManager	ManagedObjectReference:FileManager	FileManager
hostProfileManager	ManagedObjectReference:HostProfileManager	HostProfileManager
ipPoolManager	ManagedObjectReference:IpPoolManager	IpPoolManager
licenseManager	ManagedObjectReference:LicenseManager	LicenseManager
localizationManager	ManagedObjectReference:LocalizationManager	LocalizationManager
ovfManager	ManagedObjectReference:OvfManager	OvfManager

Figure 22: ServiceContent

6. Figure 23 shows the list of currently installed plug-ins on the top. Click on the “Register Extension” option. Do not worry if the current extensionList does not match; it varies from environment to environment.

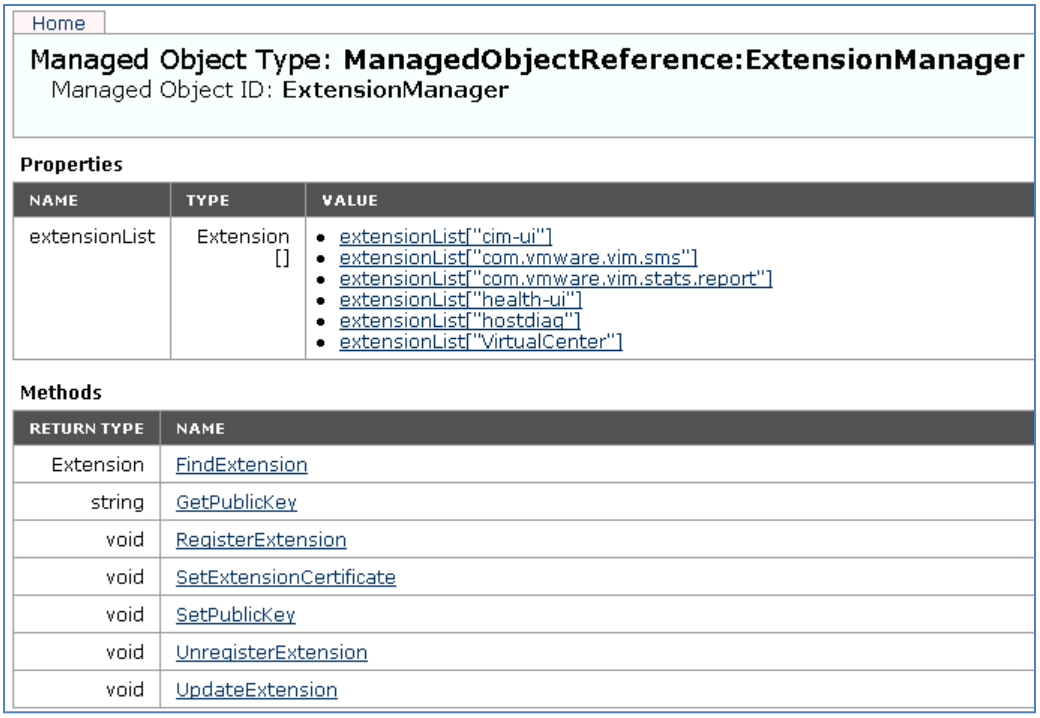


Figure 23: ExtensionManager

7. Delete the existing contents of the "value" field, and then copy the entire contents of the modified MOB_ExtMgr.xml file located at the installation folder. Click on the "Invoke Method" option. Figure 24 shows the window layout.

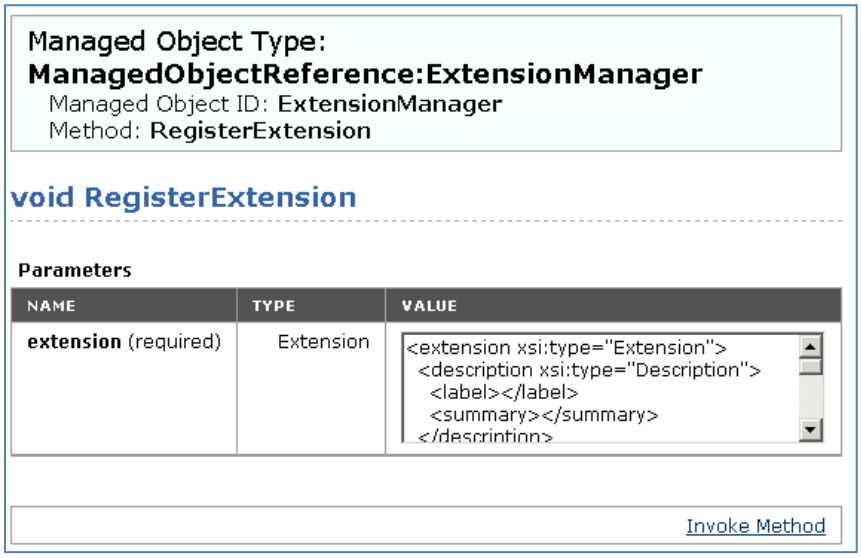
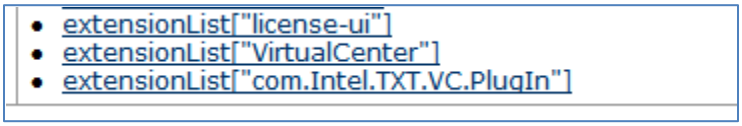


Figure 24: RegisterExtension

8. After successful registration, “Refresh” the MOB tool page showing the list of current plug-ins registered. It should show one corresponding to the Intel TXT Demo plug-in that was just registered.



Appendix B—RSA Archer Supplemental Information

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. 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 and VMWare's PowerCLI to programmatically connect, authenticate, and read the PCR data value 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.

Below is 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 <server> -port <port> -user <username> -pass <password>
$TimeStamp=$(get-date -format g)
$TpmDigestInfo= @()
Get-VMHost | Get-View | Foreach {
      $Info=New-Object PSObject
      $Info | add-member -membertype noteproperty -Name VMHost -Value $_.Name
      If ($_.Runtime.TpmPcrValues -eq$null){
            $Info | add-member -membertype noteproperty -Name TPMPcrValues -Value "Not Enabled"
      } Else {
            $Info | add-member -membertype noteproperty -Name TPMPcrValues -Value $_.Runtime.TpmPcrValues
      }
      $TpmDigestInfo+=$Info
}

$TPMSummary= @()
For ($i=0; $i -lt $tpmdigestinfo.length; $i++)
{
        $info2=new-object PSObject
        $info2 | add-member -membertype noteproperty -Name VMHost -Value $tpmdigestinfo[$i].VMHost
        $PCRstring=[string]::join("",$($TPMDigestInfo[$i].TPMPcrValues[20].DigestValue))
        $info2 | add-member -membertype noteproperty -Name PCR20string -Value $PCRstring
        $PCRstring=[string]::join("",$($TPMDigestInfo[$i].TPMPcrValues[22].DigestValue))
        $info2 | add-member -membertype noteproperty -Name PCR22string -Value $PCRstring
        $info2 | add-member -membertype noteproperty -Name TimeStamp -Value $TimeStamp
        $TPMSummary+=$info2
}

$TPMSummary | export-csv C:\Users\Administrator\Desktop\PCRData.csv -notype

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

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.

Appendix C—Demonstration Architecture

The following appendix describes the architecture used to implement the trusted cloud proof of concept demonstration.

C.1 Plug-In Installation and Configuration

The implemented architecture is composed of three Dell servers running VMware ESXi 4.1 Update 1 configured as a cluster with a shared resource pool utilizing an iSCSI storage device, a management node which includes three VMs providing different functionalities, and a dedicated management workstation.

Trusted Cloud Cluster:

- 3 x Dell PowerEdge R710 (TXT enabled):
  - 2 x Intel Xeon CPU E5645 @ 2.40GHz
  - 48 GB Memory
- VMware ESXi 4.1 Update 1 hosting the following VMs:
  - Windows Server 2008 R2 for test workload VM connected to the VM Traffic Network

Storage:

- Dell PowerVault MD3200i

Management Node:

- Dell PowerEdge R805
  - 2 x Quad-Core AMD Opteron Processor 2384
  - 32 GB Memory
- VMware ESXi 4.1 Update 1 hosting the following VM:
  - Windows Server 2008 R2 with VMware vCenter Enterprise Plus Server
  - Windows Server 2008 R2 with IIS enabled and Intel VMware Plug-in
  - Windows Server 2008 R2 with SQL Server 2008 R2 and RSA Archer eGRC

Management Workstation:

- Dell Optiplex 908
  - Windows 7 with VMware vSphere client

C.2 Setup of Components

Following the recommendations proposed in NIST SP 800-125, the architecture of the PoC implementation is composed of three distinct networks to isolate the traffic flowing through the management VMs, storage device, and public VMs.

Management Network

The Dell Optiplex management workstation is connected to the management network, which includes the four Dell servers. A dedicated ESXi server is used to host the management VMs for the vCenter VM, the VM with the Intel TXT plug-in, and the Archer VM. The vCenter VM manages the remaining 3 ESXi servers, which are part of the cluster hosting the public VMs. The VM with the Intel TXT plug-in takes

measurements of the trusted cloud cluster and directs them to the vCenter VM. The Archer VM communicates with the vCenter VM 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 vMotion which supports the automated migration of the VMs from different nodes across the trusted cluster.

Storage Network

The Dell PowerVault MD3200i iSCSI storage device provides shared storage where the public VMs are hosted. The three Dell ESXi servers are connected to the storage network, which uses a non-routable network.

Public VMs Network

The public VMs network is accessible to the workload owners from the Internet. In the demonstration, a single Windows Server 2008 R2 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 node is used to carry the VM's traffic.

Figure 25 represents the proof of concept implementation, which includes the various hardware and logical networks.

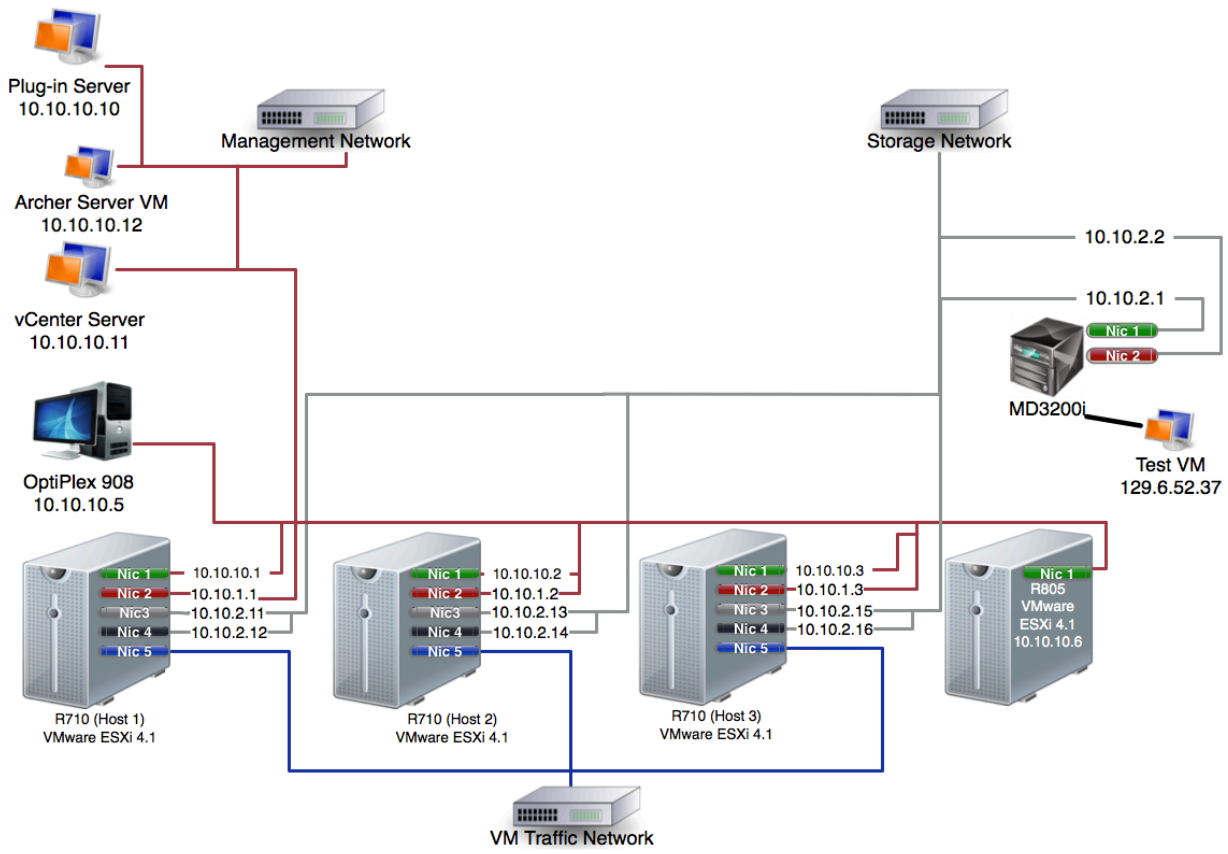


Figure 25: Proof of Concept Implementation

Appendix D—Acronyms and Other Abbreviations

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

ACM	Authenticated Code Module
AIK	Attestation Identity Key
API	Application Programming Interface
BIOS	Basic Input/Output System
CPU	Central Processing Unit
FISMA	Federal Information Security Management Act
Gb	Gigabit
GB	Gigabyte
GHz	Gigahertz
HDD	Hard Disk Drive
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
IT	Information Technology
ITL	Information Technology Laboratory
LCP	Launch Control Policy
MOB	Managed Object Browser
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
OS	Operating System
PCR	Platform Configuration Register
POC	Proof of Concept
RAM	Random Access Memory
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SDK	Software Development Kit
SP	Special Publication
TPM	Trusted Platform Module
UI	User Interface
URL	Uniform Resource Locator
VM	Virtual Machine
VMM	Virtual Machine Monitor
XML	Extensible Markup Language