Intel® Cloud Integrity Technology 2.2

Product Guide

September 2016

Intel Confidential

Revision 1.0

Revision History

Revision	Date	Comments
1.0	September 12, 2016	Initial release (Intel confidential).

Contents

Contents

1.0 Introduction

Intel® Cloud Integrity Technology 2.2 (Intel® CIT) provides visibility into the cloud data center, leveraging Intel processors with Intel® Trusted Execution Technology (Intel® TXT) to establish a root of trust and enable hardware/hypervisor-layer trust and asset tag compliance management.

1.1 Audience

This document is intended to provide guidance for installing Intel® CIT on an Ubuntu* Virtual Machine (VM) and is intended for technical personnel who are proficient in the following:

- One or more of the following hypervisors and their corresponding management stacks:
  - VMware* vSphere
  - Citrix* Xen
  - Kernel-based Virtual Machine (KVM)
  - Hyper-V 2012
- Basic networking
- Ubuntu 14.04 LTS

1.2 Overview

Intel® Cloud Integrity Technology leverages Intel® TXT to provide a root of trust in the data center, enabling software-controlled visibility and policy enforcement of hardware-secured asset tags and launch-time measurements of the BIOS, Operating System (OS), and hypervisor. The trust and asset tag attestation information can then be used to provide workload verification, remediation, reporting, and compliance in both public and private cloud environments.

1.3 The Chain of Trust

The *Chain of Trust* begins with Intel® TXT, which acts as an unchanging hardware measurement agent, referred to as a hardware *Root of Trust*. By using Intel® TXT to measure software that in turn measures other software, a chain of trust can be established, where each step of the boot process measures the next step.

Intel® Cloud Integrity Technology 2.2 extends the concept of the "chain of trust" using a remote Attestation Service to verify the measured server components against previously-stored known-good measurements, while previous incarnations were only capable of measuring the BIOS, OS kernel, and hypervisor components

1.4 Deployment Scenarios

Figure 1 shows deployment scenarios for environments consisting of ESXi, Citrix XenServer, and Trust Agent servers. Depending on the individual deployment strategy, different deployment scenarios can also be considered.

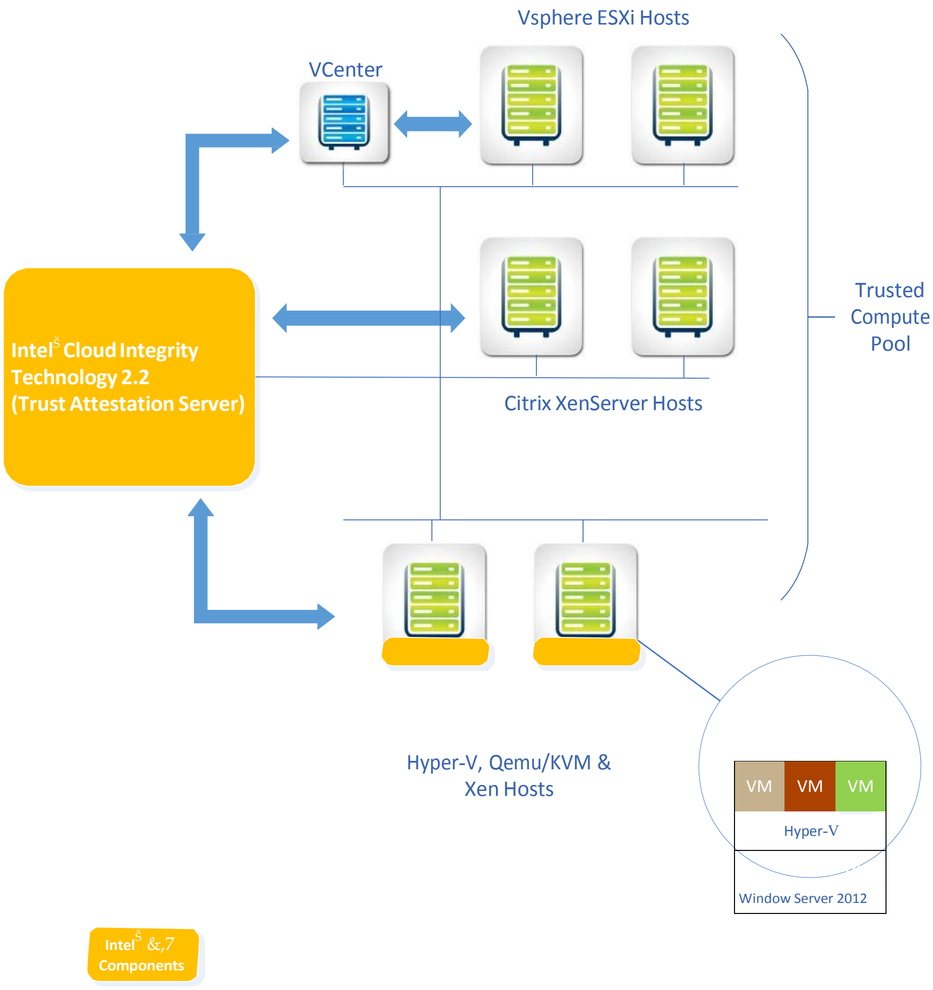
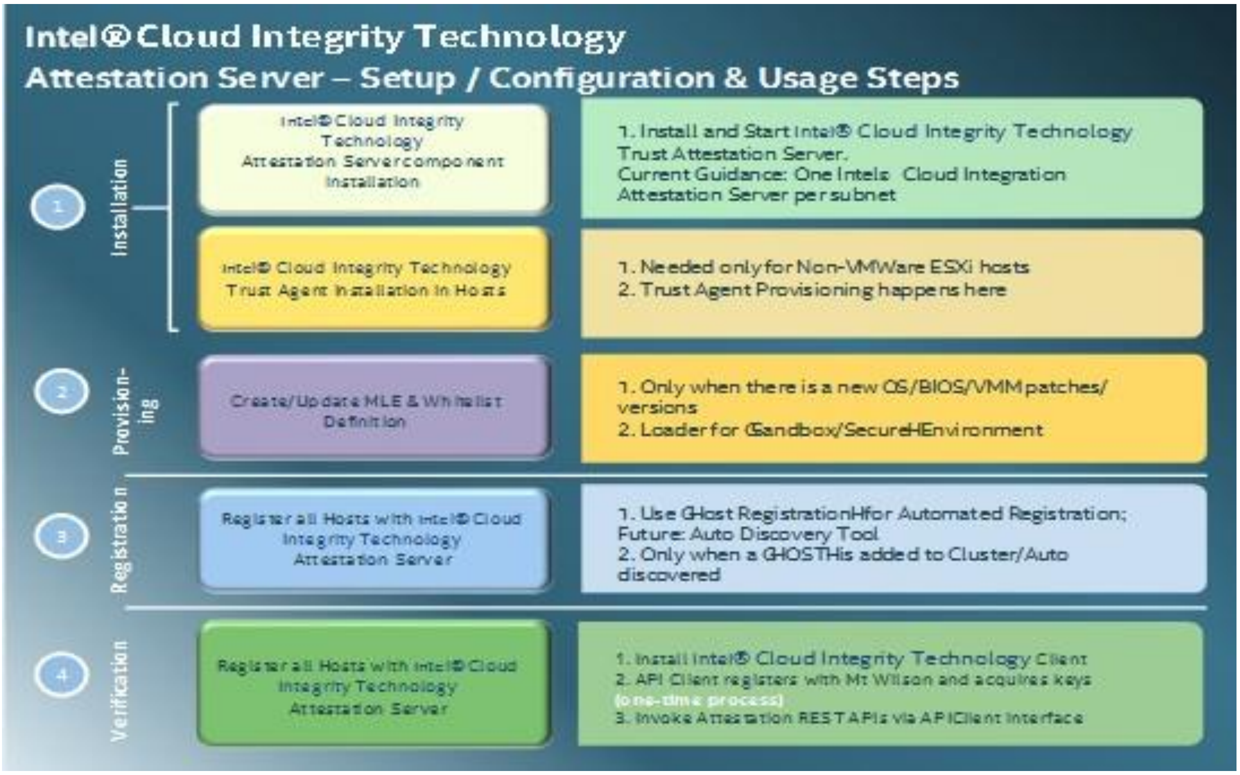


Figure 1. Deployment Scenarios

1.5 Workflow

Figure 2 shows the high-level procedure required to configure host attestation using Intel® CIT.



Intel® Cloud Integrity Technology
Attestation Server – Setup / Configuration & Usage Steps

Intel® Cloud Integrity Technology Attestation Server component Installation

1. Install and Start Intel® Cloud Integrity Technology Trust Attestation Server.
Current Guidance: One Intels Cloud Integration Attestation Server per subnet

Intel® Cloud Integrity Technology Trust Agent Installation in Hosts

1. Needed only for Non-VMWare ESXi hosts
2. Trust Agent Provisioning happens here

Create/Update MLE & Whitelist Definition

1. Only when there is a new OS/BIOS/VMM patches/ versions
2. Loader for Sandbox/SecureHEnvironment

Register all Hosts with Intel® Cloud Integrity Technology Attestation Server

1. Use GHost RegistrationHfor Automated Registration; Future: Auto Discovery Tool
2. Only when a GHOSTHis added to Cluster/Auto discovered

Register all Hosts with Intel® Cloud Integrity Technology Attestation Server

1. Install Intel® Cloud Integrity Technology Client
2. API Client registers with Mt Wilson and acquires keys (one-time process)
3. Invoke Attestation REST APIs via APIClient Interface

Figure 2. Host Attestation Configuration

Basic Workflow Steps:

1. Configure the Intel® CIT Attestation Server and its associated database.
2. Register and approve any required users with appropriate roles.
3. Install the Trust Agent on any Linux or Microsoft Windows hosts (not required for Citrix Xen or VMware environments).
4. Create BIOS and Virtual Machine Manager (VMM) Measured Launch Environments (MLEs) imported from known-good hosts to use as a basis for trust attestation.

   Note: This needs to be performed once per MLE, and should not be repeated unless a new BIOS or OS/VMM update requires the creation of a new MLE.
5. Register hosts for attestation.
6. Verify host trust status.

2.0 Intel® Cloud Integrity Technology Components

This section describes the various Intel® Cloud Integrity Technology components.

2.1 Attestation Server

The Attestation Server component of Intel® CIT performs remote attestation of physical servers, comparing Intel® TXT measurements of BIOS, OS, Asset Tag, and other components against a database of *known-good* values. The attested trust status of each server is used to make policy decisions for workload placement.

As a server boots, Intel® TXT begins extending measurements to a *Trusted Platform Module* (TPM). Each chain of trust component is measured, and these measurements are remotely verified using the Attestation Server.

Known-good measurements for each of these components can be directly imported from a sample server.

Each server to be attested is registered with the Attestation Server. This process includes setting the expected values for future attestations, and the generation of the *Attestation Identity Key* (AIK). The AIK is an asymmetric key pair generated by the host's TPM for the purpose of cryptographically securing attestation quotes for transmission to the Attestation Server.

2.2 Trust Agent

The Trust Agent resides on physical servers and enables remote attestation capabilities. The Agent maintains ownership of the server's TPM, allowing secure attestation quotes to be sent to the Attestation Service.

The Trust Agent is supported for Microsoft Windows Server, Ubuntu and Red Hat Enterprise Linux (RHEL). The Trust Agent is unnecessary for VMware or Citrix Xen deployments.

3.0 Intel® Cloud Integrity Technology Setup

This section describes the installation and configuration steps for each component of the Intel® CIT architecture.

3.1 Preparing the Build Environment

The following procedures were tested on an Ubuntu 14.04 host. Update your proxy settings if it is required.

3.1.1 Installing the Required Packages

To install the required packages, execute the following commands:

```
# cd /root
# apt-get update
# apt-get install ssh ant makeself git make gcc g++ openssl libssl-dev unzip nsis zip -y
```

3.1.2 Installing the Java Development Kit (JDK)

To install the JDK, follow these steps:

1. Download the JDK from:

   http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u51-oth-JPR

2. Download the *jdk-7u51-linux-x64.tar.gz* file.

3. Systems usually come with a default Java. You must update at least the ones used by CIT compilation. Run the following commands to set it up correctly on the system:

   ```
   # mv jdk-7u51-linux-x64.tar.gz jdk-1.7.0_51-linux-x64.tar.gz
   # gzip -dc jdk-1.7.0_51-linux-x64.tar.gz | tar xf -
   # mv jdk1.7.0_51 /usr/lib/jvm  #Create the jvm directory if it doesn’t exist already
   # cd /etc/alternatives/ #if java/javac symlinks already exist here, delete them first
   # ln -s /usr/lib/jvm/jdk1.7.0_51/bin/java  # ln -s /usr/lib/jvm/jdk1.7.0_51/bin/javac
   # cd /usr/bin/  #if java/javac symlinks already exist here, delete them first
   # ln -s /etc/alternatives/java
   # ln -s etc/alternatives/javac
   ```

3.1.3 Installing Apache Maven 3.3.3

To install Apache Maven 3.3.3, follow these steps:

1. Download the binary from repository with the following command:

   ```
   # wget http://archive.apache.org/dist/maven/maven-3/3.3.1/binaries/apache-
   maven-3.3.1-bin.zip
   ```

2. Unzip the file.

   ```
   # unzip apache-maven-3.3.1-bin.zip
   ```

3. Move the application directory to */usr/local*.

   ```
   # mv apache-maven-3.3.1 /usr/local
   ```

3.1.4 Editing *setting.xml*

If you need proxy settings, do so in the */usr/local/apache-maven-3.3.1/conf/settings.xml* file. It is located in the *conf* directory of your maven app directory:

```
<proxy>
    <id>YourProxyId</id>
    <active>true</active>
    <protocol>http</protocol>
    <host>YourProxyIP</host>
    <port>YourProxyPort</port>
</proxy>
```

3.1.5 Modifying Environment Files

To modify the environment files, follow these steps:

1. Add environment variables to *~/.bashrc*.

```
JAVA_HOME=/usr/lib/jvm/jdk1.7.0_51
export JAVA_HOME
PATH=$PATH:$JAVA_HOME
export M2_HOME=/usr/local/apache-maven-3.3.1
export M2=$M2_HOME/bin
export MAVEN_OPTS="-Xmx1024m -XX:MaxPermSize=1024m"
PATH=$PATH:$M2
```

2. Restart your session so the environment variables are loaded correctly.

3. Once you login again, verify maven installation:

```
Mvn-version
```

3.2 Intel® CIT Source Code

3.2.1 Downloading the Source Code

The source code must be downloaded at */root*. When you clone, the default branch is master. Use this if the latest published code is required.

```
# cd /root
# git clone https://github.com/opencit/opencit-external-artifacts/
# git clone https://github.com/opencit/opencit-util/
# git clone https://github.com/opencit/opencit/
```

3.2.2 Building the Source Code

The source code must be built in the following order:

1. External Artifacts [*opencit-external-artifacts*]

2. Util Project [*opencit-util*]

3. Open CIT [*opencit*]

Note: External Artifacts requires additional steps to build. See the *Readme.md* file for more information on how to build this project.

To build the source code, follow these steps:

1. Run **ant** to build each project.

```
# cd /root/opencit-external-artifacts
# ant
# cd ..
# cd /root/opencit-util
# git checkout release-cit-2.2-beta
# ant
# cd ..
# cd /root/opencit
# git checkout release-cit-2.2-beta
# ant
```

2. Verify that all 3 builds are successful.

3.3 Building External Artifacts

The Open CIT external artifacts project provides has several external dependencies required to successfully build Open CIT.

3.3.1 Prerequisites

To build this project, you must download a couple of artifacts and place them in specific directories so we can help you install to your local maven repo.

1. After you clone the project, **cd** to the *root* directory of external artifacts.

```
# cd opencit-external-artifacts
# git checkout release-cit-2.2-beta
```

2. Download each artifact and prepare them to build the *opencit-external-artifacts* project.

Monit

Execute the following commands:

```
# wget https://mmonit.com/monit/dist/monit-5.5.tar.gz
# mv monit-5.5.tar.gz monit/monit-5.5-linux-src.tgz
```

Tomcat

Execute the following commands:

```
# wget https://archive.apache.org/dist/tomcat/tomcat-7/v7.0.34/bin/apache-tomcat-
7.0.34.tar.gz   mv apache-tomcat-7.0.34.tar.gz apache-tomcat/apache-tomcat-
7.0.34.tar.gz
```

Glassfish

Execute the following commands:

```
# wget http://download.java.net/glassfish/4.0/release/glassfish-4.0.zip
# mv glassfish-4.0.zip glassfish/glassfish-4.0.zip
```

vijava

Execute the following commands:

```
# wget https://sourceforge.net/projects/vijava/files/vijava/
VI%20Java%20API%205.5%20Beta/vijava55b20130927.zip
# unzip vijava55b20130927.zip
# mv vijava55b20130927.jar vijava/vijava-5.5.jar
```

JDK

1. Use the JDK downloaded in Section 3.1.2. If you do not have it, you can download it from:

   http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jdk-7u51-oth-JPR

2. Download the *jdk-7u51-linux-x64.tar.gz* file.
3. Once you have the file, place it in the *jdk* directory of the *opencit-external-artifacts* project

   ```
   # mv jdk-7u51-linux-x64.tar.gz jdk/jdk-1.7.0_51-linux-x64.tar.gz
   ```

4. Now clean up.

   ```
   # rm -rf *.zip
   # rm -rf *.jar
   # rm -rf XenServer-SDK
   ```

3.3.2 Build

Build using:

```
# ant
```

3.3.3 Binary locations

A list of significant binary artifacts can be found in the top of the repository in a file called "build.targets"

3.4 Installing the Attestation Server

The Intel® CIT Attestation Server can be run as a VM or as a bare-metal server.

3.4.1 Package Dependencies

The Intel® CIT Attestation Service requires the following packages:

- Monit (optional)
- Logback (optional)
- Apache Tomcat
- Java 7u51 JDK
- OpenSSL
- Postgres client and server 9.3
- Unzip

If they are not already installed, the Attestation Service installer attempts to install these packages automatically using the package manager. Automatic installation requires access to package repositories, which may require an Internet connection. If the packages are to be installed from the package repository, be sure to update your repository package lists before installation.

3.4.2 Supported Operating Systems

- Ubuntu 14.04

3.4.3 Recommended Hardware

- 2 vCPUs
- RAM: 4 GB Recommended; 2 GB Minimum
- 2 GB free space to install the Attestation Server services. Additional free space may be needed if the Attestation Server database is located on the same disk (database space requirement is dependent on the number of managed servers and the frequency with which they are attested).
- One network interface with network access to all managed servers.
- (Optional) One network interface for Asset Tag provisioning (only required for "pull" tag provisioning; required to provision Asset Tags to VMware ESXi servers).

3.4.4 Installation

To install the Attestation Service, follow these steps:

1. Copy the Attestation Service installation binary to the */root/* directory.
2. Create the *mtwilson.env* installation answer file for an unattended installation.

   A sample minimal *mtwilson.env* file is provided next. For all configuration options and their descriptions, refer to Section 5.1.

```
export INSTALL_PKGS="java logrotate tomcat postgres privacyca monit SERVICES PORTALS"
export MTWILSON_SERVER=<IP address or hostname for the Attestation Server>
export MTWILSON_API_BASEURL="https://${MTWILSON_SERVER}:8443/mtwilson/v1"
export WEBSERVER_VENDOR=tomcat
export DATABASE_VENDOR=postgres
export DATABASE_PORTNUM=5432
export MC_FIRST_USERNAME=administrator
export MC_FIRST_PASSWORD=password
export TOMCAT_ADMIN_USERNAME=tomcatadmin
export TOMCAT_ADMIN_PASSWORD=setininstall
export PRIVACYCA_DOWNLOAD_USERNAME=pcaadmin
export  PRIVACYCA_DOWNLOAD_PASSWORD=password
######################################################################################################
export MTWILSON_TLS_POLICY_ALLOW=certificate,certificate-digest,public-key,public-key-digest,TRUST_FIRST_CERTIFICATE
export MTWILSON_DEFAULT_TLS_POLICY_ID=TRUST_FIRST_CERTIFICATE
######################################################################################################
export LOG_ROTATION_PERIOD=daily
export LOG_COMPRESS=compress
export LOG_DELAYCOMPRESS=delaycompress
export LOG_COPYTRUNCATE=copytruncate
export LOG_SIZE=100M
export LOG_OLD=7
######################################################################################################
export JAVA_REQUIRED_VERSION=1.7.0_51
export MTWILSON_API_SSL_VERIFY_HOSTNAME=false
export DATABASE_HOSTNAME=127.0.0.1
export DATABASE_SCHEMA=mw_as
export DATABASE_USERNAME=root
export DATABASE_PASSWORD=password
######################################################################################################
export POSTGRES_REQUIRED_VERSION=9.3
export POSTGRES_HOSTNAME=${DATABASE_HOSTNAME}
export POSTGRES_PORTNUM=${DATABASE_PORTNUM}
export POSTGRES_DATABASE=${DATABASE_SCHEMA}
export POSTGRES_USERNAME=${DATABASE_USERNAME}
export  POSTGRES_PASSWORD=${DATABASE_PASSWORD}
export ADD_POSTGRESQL_REPO=yes
export POSTGRESQL_KEEP_PGPASS=true
######################################################################################################
export TOMCAT_REQUIRED_VERSION=7.0
export TOMCAT_CREATE_SSL_CERT=yes
export TOMCAT_SSL_CERT_CN=${MTWILSON_SERVER},127.0.0.1
export PRIVACYCA_SERVER=${MTWILSON_SERVER}
export INSTALLED_MARKER_FILE=/var/opt/intel/.mtwilsonInstalled
export INSTALL_LOG_FILE=/tmp/mtwilson-install.log
######################################################################################################
# ASSET TAG Configuration
export MTWILSON_TAG_SERVER_PRIVATE=${MTWILSON_SERVER}
export TAG_PROVISION_EXTERNAL=false export TAG_PROVISION_NOCACHE=true
export TAG_PROVISION_XML_ENCRYPTION_REQUIRED=false
export TAG_PROVISION_XML_PASSWORD=password
export MTWILSON_TAG_KEYSTORE="/opt/mtwilson/configuration/serverAtag.jks"
exportMTWILSON_TAG_KEYSTORE_PASSWORD=password
export MTWILSON_TAG_KEY_PASSWORD=password
export MTWILSON_TAG_CERT_IMPORT_AUTO=true
export TAG_PROVISION_XML_REQUIRED=false
export export TAG_VALIDITY_SECONDS=31536000
export TAG_ISSUER_DN="CN=mtwilson-tag-ca"
export MTWILSON_TAG_HTML5_DIR="/opt/mtwilson/share/apache-tomcat-7.0.34/webapps/mtwilson-portal/tag"
export MTWILSON_TAG_ADMIN_USERNAME=tagadmin
export MTWILSON_TAG_ADMIN_PASSWORD=password
```

3. Execute the installer binary.

When the installation completes, the Attestation Service is running. If the web portal component was installed (recommended), the portal UI is now accessible at the following URL:

https://<*Attestation_Server_IP_Address*>:8443/mtwilson-portal/index.html

The services can also be verified by running **mtwilson status** from the Attestation Service command line.

3.5 Installing the Trust Agent

Installation of the Intel® CIT Trust Agent differs for Linux and Windows hosts.

3.5.1 Installing the Trust Agent - Linux (TPM 1.2)

3.5.1.1 Package Dependencies

The Intel® CIT Linux Trust Agent for requires the following packages:

- trousers
- patched-tpm-tools
- openssl
- make
- gcc

If they are not already installed, the Trust Agent installer attempts to install these automatically using the package manager. Automatic installation requires access to package repositories, which may require an Internet connection. If the packages are to be installed from the package repository, be sure to update the repository package lists before installation.

3.5.1.2 Supported Operating Systems

The Intel® CIT Linux Trust Agent for TPM 1.2 supports the following operating systems:

- Ubuntu 14.04
- Red Hat Enterprise Linux 7.2

3.5.1.3 Prerequisites

The following must be completed before installing the Trust Agent:

- Supported server hardware including an Intel® Xeon® processor with Intel® Trusted Execution Technology activated in the system BIOS.
- Trusted Platform Module (TPM 1.2) installed and activated in the system BIOS, with cleared ownership status.
- System must be booted to a tboot 1.8.1 or higher boot option with Trousers and the Intel® CIT patched version of the TPM Tools package installed (for Trusted Boot provisioning steps, refer to Section 8.1).
- Intel-provided *patched-tpm-tools* package must be installed (for Trusted Boot provisioning steps, refer to Section 8.1).
- Intel® CIT Attestation Service server installed and active.

3.5.1.4 Installation

To install the Intel® CIT Linux Trust Agent:

1. Create the *trustagent.env* answer file in the */root/* directory (for configuration options, see Section 5.9.2).

```
MTWILSON_API_URL=https://<Attestation Service IP or Hostname>:8443/mtwilson/v2
MTWILSON_API_USERNAME=<Attestation Service PrivacyCA username>
MTWILSON_API_PASSWORD=<Attestation Service PrivacyCA password>
MTWILSON_TLS_CERT_SHA256=<SHA256 of Attestation Service TLS Certificate>
REGISTER_TPM_PASSWORD=y
TRUSTAGENT_LOGIN_REGISTER=true
```

   Note: The MTWILSON_API_USERNAME and password required by the Trust Agent can be satisfied by the PRIVACYCA_DOWNLOAD_USERNAME user created during the installation of the Attestation Service.

2. Copy the Trust Agent installer binary to the */root/* directory.

   Note: There are separate Trust Agent installers for Ubuntu and Red Hat. Select the installer appropriate for your OS.

3. Execute the Trust Agent installer, and wait for the installation to complete.

   When the installer completes, the Trust Agent will be running.

3.5.2 Installing the Trust Agent - Linux (TPM 2.0)

3.5.2.1 Package Dependencies

The Intel® CIT Linux Trust Agent for requires the following packages:

- TSS 2 packages
- openssl
- make
- gcc

The trousers and tpm2-tools are installed with the Trust Agent installer. Do not download these from the distro. Automatic installation requires access to package repositories, which may require an Internet connection.

3.5.2.2 Supported Operating Systems

The Intel® CIT Linux Trust Agent for TPM 2.0 supports the following operating systems:

- Ubuntu 16.04 LTS
- Red Hat Enterprise Linux 7.2

3.5.2.3 Prerequisites

The following must be completed before installing the Trust Agent:

- Supported server hardware including an Intel® Xeon® processor with Intel® Trusted Execution Technology activated in the system BIOS.
- Trusted Platform Module (TPM 2.0) installed and activated in the system BIOS, with cleared ownership status.

- Intel-provided tboot_1.9.4 or higher with TPM 2.0 driver already enabled.
- Intel® CIT Attestation Service server installed and active.

3.5.2.4 Installation

To install the Intel® CIT Linux Trust Agent:

1. Create the *trustagent.env* answer file in the */root/* directory (for configuration options, see Section 5.9.2).

```
MTWILSON_API_URL=https://<Attestation Service IP or Hostname>:8443/mtwilson/v2
MTWILSON_API_USERNAME=<Attestation Service PrivacyCA username>
MTWILSON_API_PASSWORD=<Attestation Service PrivacyCA password>
MTWILSON_TLS_CERT_SHA256=<SHA256 of Attestation Service TLS Certificate>
REGISTER_TPM_PASSWORD=y
TRUSTAGENT_LOGIN_REGISTER=true
```

   Note: The MTWILSON_API_USERNAME and password required by the Trust Agent can be satisfied by the PRIVACYCA_DOWNLOAD_USERNAME user created during the installation of the Attestation Service.

2. Copy the Trust Agent installer binary to the */root/* directory.

   Note: There are separate Trust Agent installers for Ubuntu and Red Hat. Select the installer appropriate for your OS.

3. Execute the Trust Agent installer, and wait for the installation to complete.

   When the installer completes, the Trust Agent will be running.

3.5.3 Installing the Trust Agent - Windows (TPM 1.2/2.0)

3.5.3.1 Package Dependencies

The Intel® CIT Windows Trust Agent has the following software dependencies:

- Microsoft Visual C++ Redistributable
- Java JDK 7u51

If these are not already installed, the Trust Agent installer attempts to install them, which may require an Internet connection.

3.5.3.2 Supported Operating Systems

The Intel® CIT Windows Trust Agent supports the following operating systems:

- Microsoft Windows Server 2012
- Microsoft Hyper-V 2012

3.5.3.3 Prerequisites

The following must be completed before installing the Trust Agent:

- Supported server hardware including an Intel® Xeon® processor with Intel® Trusted Execution Technology activated in the system BIOS.
- Trusted Platform Module installed and activated in the system BIOS, with cleared ownership status.

- Intel® CIT Attestation Service server installed and active.

3.5.3.4 Installation

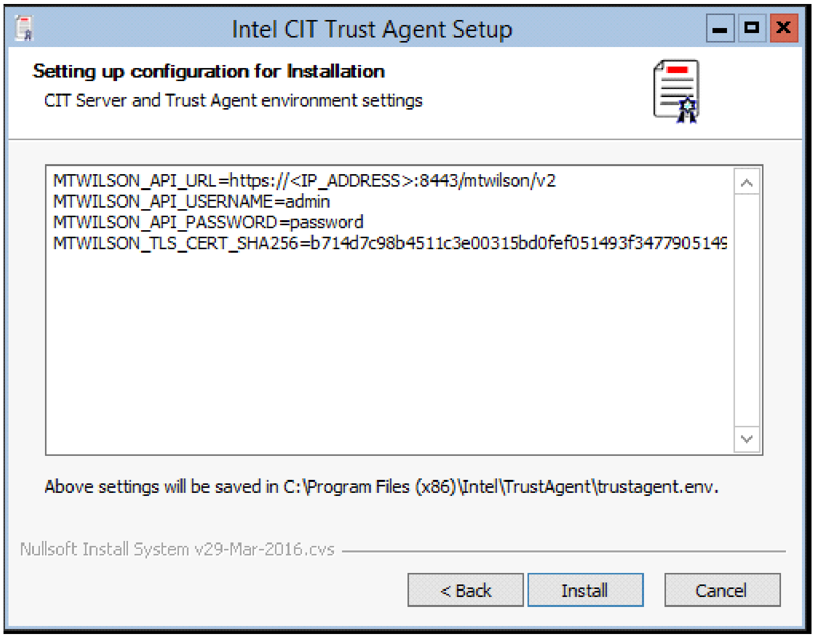
To install the Intel® CIT Windows Trust Agent:

1. Execute the Trust Agent Windows installer.
2. Copy the table of configuration settings from the table below, and enter the appropriate settings: (for configuration options, see Section 5.10.2)

```
MTWILSON_API_URL=https://<Attestation Service IP or Hostname>:8443/mtwilson/v2
MTWILSON_API_USERNAME=<Attestation Service PrivacyCA username>
MTWILSON_API_PASSWORD=<Attestation Service PrivacyCA password>
MTWILSON_TLS_CERT_SHA256=<SHA256 of Attestation Service TLS Certificate>
REGISTER_TPM_PASSWORD=y
TRUSTAGENT_LOGIN_REGISTER=true
```

Note: The MTWILSON_API_USERNAME and password required by the Trust Agent can be satisfied by the PRIVACYCA_DOWNLOAD_USERNAME user created during the installation of the Attestation Service.

3. Copy the configuration settings, and paste them into the Trust Agent installer window.



Setting up configuration for Installation

CIT Server and Trust Agent environment settings

```
MTWILSON_API_URL=https://<IP_ADDRESS>:8443/mtwilson/v2
MTWILSON_API_USERNAME=admin
MTWILSON_API_PASSWORD=password
MTWILSON_TLS_CERT_SHA256=b714d7c98b4511c3e00315bd0fef051493f3477905149
```

Above settings will be saved in C:\Program Files (x86)\Intel\TrustAgent\trustagent.env.

Nullsoft Install System v29-Mar-2016.cvs

4. Click **Install** to install the Trust Agent.

When the installation completes, the Trust Agent will be running in the background.

4.0 Getting Started

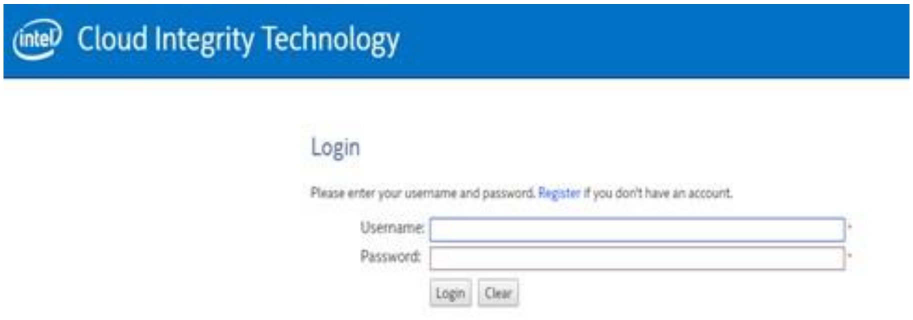
4.1 Portal Overview

The Intel® CIT Attestation Server web portal is made up of several tabs:

- **Trust** — This tab contains functions related to trust assertion. The Trust Dashboard is the home page for the web portal, and lists the trust status of all hosts currently registered in the Intel® CIT attestation database. Other functions include the Bulk Refresh tool that enables the trust status of multiple hosts to be refreshed simultaneously along with the Reports tool that contains a history of trust refreshes done on all registered hosts.
- **Host Management** — This tab contains functions related to registering, editing, and deleting hosts in the Intel® CIT Attestation Server. The Import tool allows for the automated registration of one or more hosts from a text file or a vCenter cluster. Note that whitelist MLEs must already exist before hosts can be added. The Add Host and Edit Host tools enable hosts to be manually, individually added or edited (for example, the Edit Host tool allows the MLEs used for trust attestation of a particular host to be changed in the event of a BIOS update). The View Host tool displays a list of all registered hosts and their relevant details, including MLE names, connection strings, and so on.
- **Whitelist** — This tab contains functions relating to MLE management. The Import from Trusted Host tool enables MLEs to be automatically imported from a known-good host. Note that the MLE values are taken from the known-good host and stored in the database for attestation. The known-good host does not need to actually be registered for trust attestation, and if its own module/PCR values change and no longer matches the imported MLEs, it is flagged as un-trusted). Other tools include the Edit MLE, Edit OS, and Edit OEM tools, which allow MLEs, operating systems, and OEMs to be deleted and altered. Note that an OEM, OS, or MLE can only be deleted if no currently-registered hosts are using them.
- **Asset Tag Management** — This tab contains functions surrounding creating and deploying asset tags. Functions include creating key/value pairs, creating selections of key/value pairs, and deploying selections to target hosts registered in the Intel® CIT Attestation Server.
- **Administration** — This tab contains functions related to the administration of user accounts, the Intel® CIT Attestation Server certificates, and TLS policy management. Functions include viewing all user requests, approving users (and defining their roles), deleting users, extending the expiration of user accounts, creating/editing shared TLS policies, and downloading the various certificates used by Intel® CIT.

These tabs and their included functions are described in detail in the sections that follow. The portal URL is:

https://<*Attestation_Server_IP_Address*>:8443/mtwilson-portal



Cloud Integrity Technology

Login

Please enter your username and password. Register if you don't have an account.

Figure 3. Intel® Cloud Integrity Technology Login Screen

4.2 Whitelist

A *whitelist* consists of a series of PCR measurements that are considered to be known-good values for various host platforms. These values are then used as the basis of comparison for trust attestation. Before a host can be registered, valid MLE values must already exist for the host's BIOS and VMM. Once these known-good values are created, they remain unchanged unless overwritten or deleted by user action (such as they are not specifically tied to any specific host after creation).

See Section 5.1 for details on the behavior of the Platform Configuration Registers (PCRs) that make up the MLEs and whitelist management.

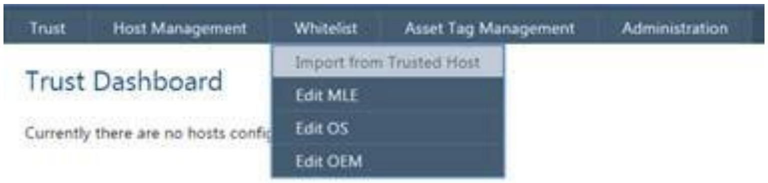
4.2.1 Importing Whitelist MLEs

MLEs are imported directly from known-good hosts. This process also sets the attestation policy. When a PCR is selected while creating an MLE, the attestation policy is also being set to attest against that PCR. While specific PCR values can be overwritten by re-importing the same MLE, this policy setting can only be changed by deleting the MLE entirely and then re-creating it. Similarly, a PCR cannot be removed from an MLE simply be overwriting it with the PCR de-selected; the MLE must be deleted and re-created without the undesired PCR selected.

4.2.2 Importing Whitelist MLE Values from a Windows Trust Agent Host

To import whitelist MLEs from an existing server:

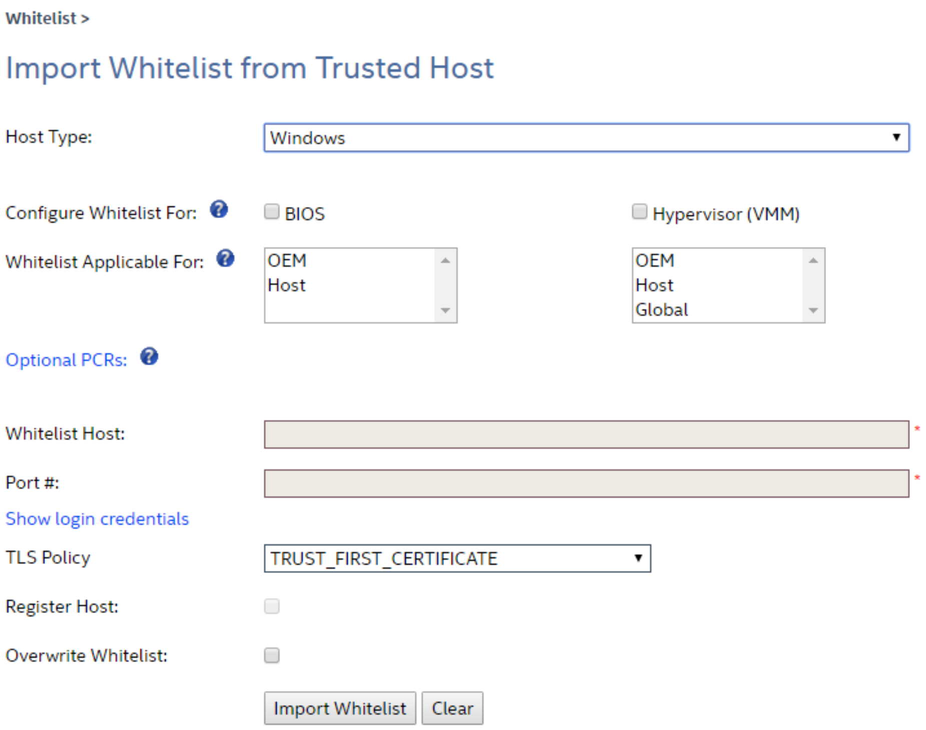
1. Browse to the Attestation Server URL, log on, and select **Whitelist > Import from Trusted Host**.



Trust Dashboard

Currently there are no hosts confic

The **Import Whitelist from Trusted Host** screen displays.



Whitelist >

Import Whitelist from Trusted Host

Host Type: Windows ▾

Configure Whitelist For: □ BIOS □ Hypervisor (VMM)

Whitelist Applicable For: OEM, Host | OEM, Host, Global

Optional PCRs:

Whitelist Host: *

Port #: *

Show login credentials

TLS Policy: TRUST_FIRST_CERTIFICATE ▾

Register Host: □

Overwrite Whitelist: □

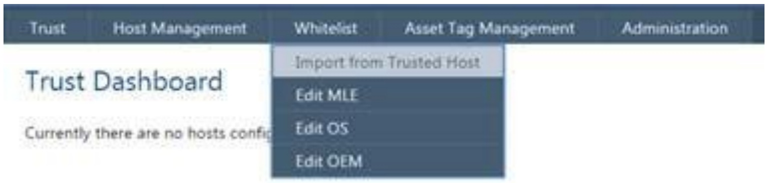
2. Select the **Host Type** as **Windows**.
3. To configure the whitelist for BIOS/VMM on the host, select the **BIOS** and **Hypervisor (VMM)** check boxes.
4. The Attestation Server automatically selects PCR 0, 13 and 14. Other PCRs can be optionally selected.

   Note: Refer to Section 5.3, "Whitelisting Guidelines" for an explanation of different whitelist types and PCR behaviors. Note: Windows, ESXi and Citrix default selections are different.
5. Enter the IP Address or DNS name of the host to be added to the whitelist.
6. Enter **1443** in the **Port #** field.
7. Select an existing shared **TLS Policy**, or create a new host-specific TLS policy for use with this host (for details, refer to Section 5.5, "TLS Policy Overview").
8. Check the box in the **Register Host** field to register this host after the whitelist MLE values are imported. Otherwise, leave unchecked.
9. Check the box in the **Overwrite Whitelist** field to overwrite an existing MLE for the same BIOS version and OS/hypervisor combination. Otherwise, leave unchecked.
10. Click **Import Whitelist** to import the MLE values from the designated host.

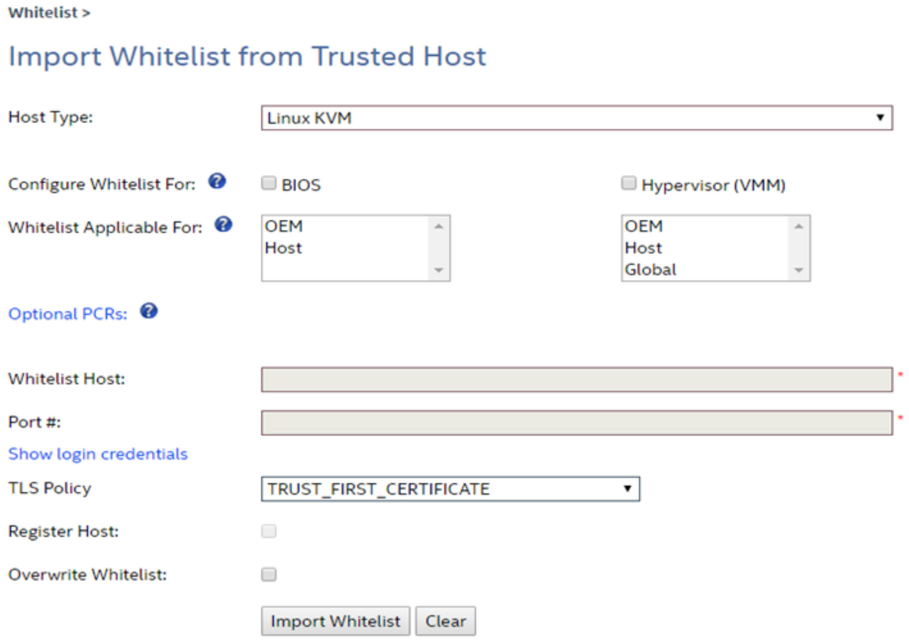
4.2.3 Importing Whitelist MLE Values from a KVM or Xen Trust Agent Host

To import whitelist MLEs from an existing server:

1. Browse to the Attestation Server URL, log on, and select **Whitelist > Import from Trusted Host**.



The **Import Whitelist from Trusted Host** screen displays.


Whitelist >

Import Whitelist from Trusted Host

Host Type: Linux KVM

Configure Whitelist For: □ BIOS □ Hypervisor (VMM)

Whitelist Applicable For: OEM, Host (BIOS); OEM, Host, Global (Hypervisor (VMM))

Optional PCRs:

Whitelist Host: *

Port #: *

Show login credentials

TLS Policy: TRUST_FIRST_CERTIFICATE

Register Host: □

Overwrite Whitelist: □

2. Select the **Host Type** as **KVM** or **Xen**.
3. To configure the whitelist for BIOS/VMM on the host, select the **BIOS** and **Hypervisor (VMM)** check boxes.
4. The Attestation Server automatically selects PCR 0, 17and 18 for Linux hosts. Other PCRs can be optionally selected. Note: Windows, ESXi and Citrix default selections are different.

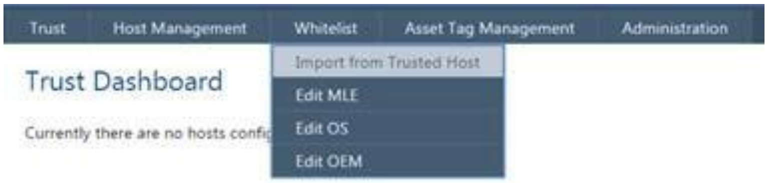
   Note: Refer to Section 5.3, "Whitelisting Guidelines" for an explanation of different whitelist types and PCR behaviors.
5. Enter the IP Address or DNS name of the host to be added to the whitelist.
6. Enter **1443** in the **Port #** field.
7. Select an existing shared **TLS Policy**, or create a new host-specific TLS policy for use with this host (for details, refer to Section 5.5, "TLS Policy Overview").
8. Check the box in the **Register Host** field to register this host after the whitelist MLE values are imported. Otherwise, leave unchecked.
9. Check the box in the **Overwrite Whitelist** field to overwrite an existing MLE for the same BIOS version and OS/hypervisor combination. Otherwise, leave unchecked.

   Note: See Section 5.4 for details on MLE management.
10. Click **Import Whitelist** to import the MLE values from the designated host.

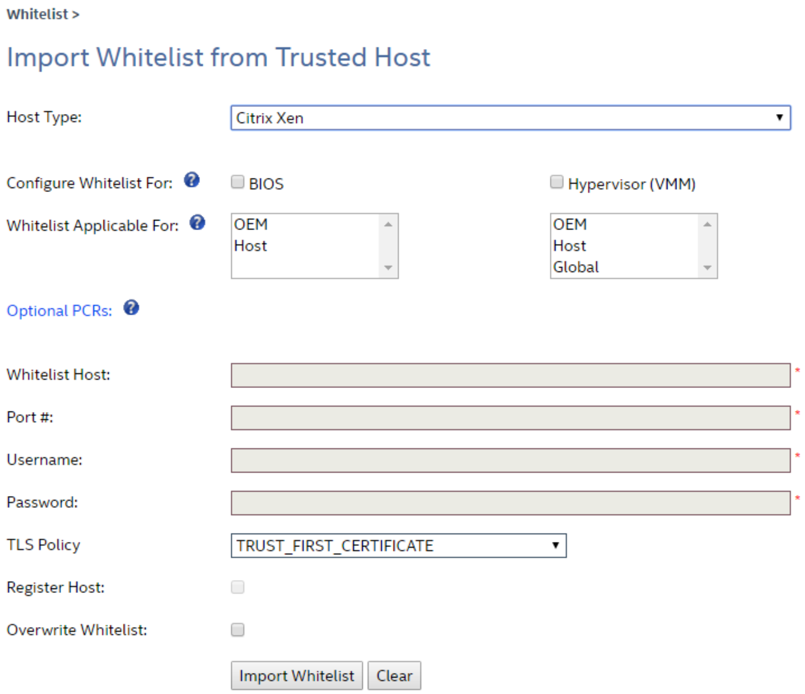
4.2.4 Importing Whitelist MLE Values from a Citrix Xen Trust Agent Host

To import whitelist MLEs from an existing server:

1. Browse to the Attestation Server URL, log on, and select **Whitelist > Import from Trusted Host**.



The **Import Whitelist from Trusted Host** screen displays.



Whitelist >

Import Whitelist from Trusted Host

Host Type: Citrix Xen

Configure Whitelist For: □ BIOS □ Hypervisor (VMM)

Whitelist Applicable For: OEM, Host | OEM, Host, Global

Optional PCRs:

Whitelist Host: *

Port #: *

Username: *

Password: *

TLS Policy: TRUST_FIRST_CERTIFICATE

Register Host: □

Overwrite Whitelist: □

2. Select the **Host Type** as **Citrix XenServer**.
3. To configure the whitelist for BIOS/VMM on the host, select the **BIOS** and **Hypervisor (VMM)** check boxes.
4. The Attestation Server automatically selects PCR 0, 17 and 18 for Citrix hosts. Other PCRs can be optionally selected. Note: Windows, ESXi and Citrix default selections are different.

   Note: Refer to Section 5.3, "Whitelisting Guidelines" for an explanation of different whitelist types and PCR behaviors.

5. Enter the IP Address or DNS name of the host to be added to the whitelist.
6. Enter **1443** in the **Port #** field.
7. Enter the username and password in the **Username** and **Password** fields, respectively.

8. Select an existing shared **TLS Policy**, or create a new host-specific TLS policy for use with this host (for details, refer to Section 5.5, "TLS Policy Overview").
9. Check the box in the **Register Host** field to register this host after the whitelist MLE values are imported. Otherwise, leave unchecked.
10. Check the box in the **Overwrite Whitelist** field to overwrite an existing MLE for the same BIOS version and OS/hypervisor combination. Otherwise, leave unchecked.

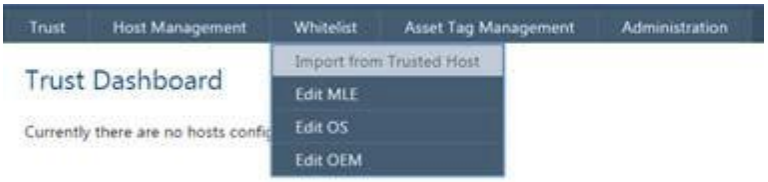
    Note: See Section 5.4 for details on MLE management.

11. Click **Import Whitelist** to import the MLE values from the designated host.

4.2.5 Importing Whitelist MLE Values from an ESXi Host

To import whitelist MLEs from an existing server:

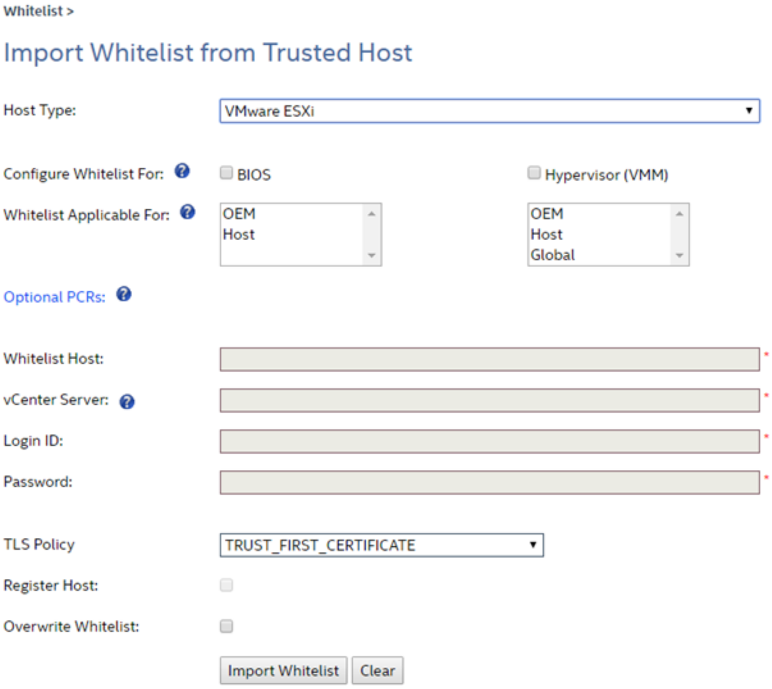
1. Browse to the Attestation Server URL, log on, and select **Whitelist > Import from Trusted Host**.



Trust Dashboard

Currently there are no hosts config

The **Import Whitelist from Trusted Host** screen displays.



Whitelist >

Import Whitelist from Trusted Host

Host Type: VMware ESXi

Configure Whitelist For: ☐ BIOS ☐ Hypervisor (VMM)

Whitelist Applicable For: OEM, Host | OEM, Host, Global

Optional PCRs:

Whitelist Host: *

vCenter Server: *

Login ID: *

Password: *

TLS Policy TRUST_FIRST_CERTIFICATE

Register Host: ☐

Overwrite Whitelist: ☐

2. Select the **Host Type** as **VMware ESXi**.

3. To configure the whitelist for BIOS/VMM on the host, select the **BIOS** and **Hypervisor (VMM)** check boxes.
4. The Attestation Server automatically selects PCR 0, 17, 18, 19 and 20. Other PCRs can be optionally selected. Other PCRs can be optionally selected.

   Note: Refer to Section 5.3, "Whitelisting Guidelines" for an explanation of different whitelist types and PCR behaviors. Note: Windows, ESXi and Citrix default selections are different.

5. In the **Whitelist Host** field, enter the name of a good known host as it appears in vCenter in the *Good Known Host* field.

   Note: The name can be the host's DNS name or IP Address but must match what is shown in vCenter.

6. In the **vCenter Server** field, enter the IP Address or host name of the vCenter server.
7. Enter the **Login ID** and **Password** for an account with administrator rights to the vCenter.
8. Select an existing shared **TLS Policy**, or create a new host-specific TLS policy for use with this host (for details, refer to Section 5.5, "TLS Policy Overview").
9. Check the box in the **Register Host** field to register this host after the whitelist MLE values are imported. Otherwise, leave unchecked.
10. Check the box in the **Overwrite Whitelist** field to overwrite an existing MLE for the same BIOS version and OS/hypervisor combination. Otherwise, leave unchecked.

    Note: See Section 5.4 for details on MLE management.

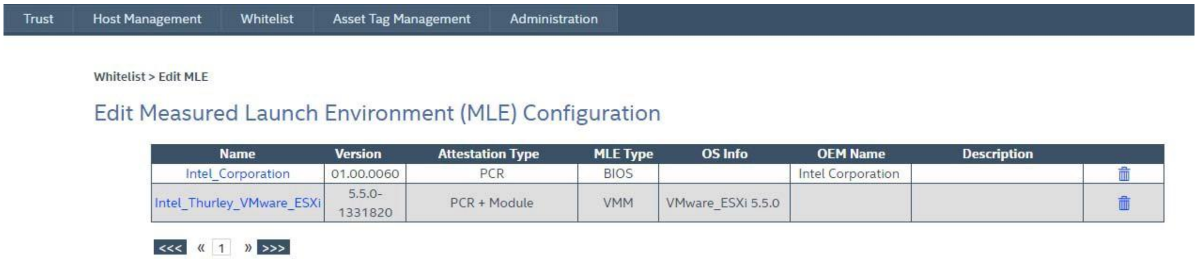
11. Click **Import Whitelist** to import the MLE values from the designated host.

4.2.6 Edit/View MLE

This tool enables specific MLEs to be managed and deleted.

Select **Whitelist > Edit MLE**.

The default view shows all MLEs (both BIOS and VMM) currently in the Intel® CIT Attestation Server database.



Whitelist > Edit MLE

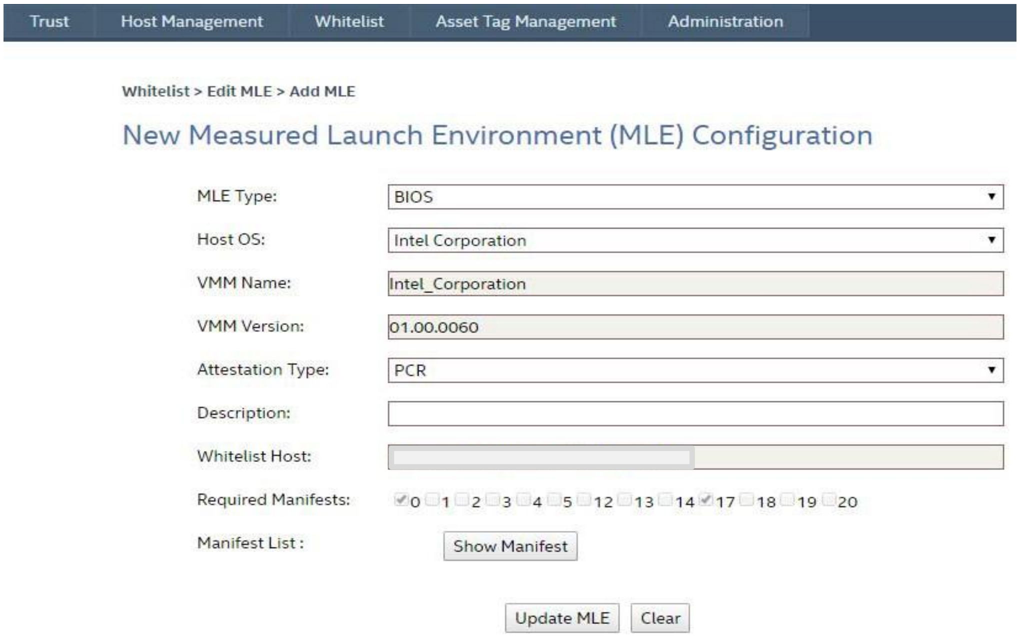
Edit Measured Launch Environment (MLE) Configuration

Name	Version	Attestation Type	MLE Type	OS Info	OEM Name	Description	
Intel_Corporation	01.00.0060	PCR	BIOS		Intel Corporation		🗑
Intel_Thurley_VMware_ESXi	5.5.0-1331820	PCR + Module	VMM	VMware_ESXi 5.5.0			🗑

To view the details for a specific MLE, click the BIOS or VMM MLE Name hyperlink.

The MLE details are listed for the specific MLE Type. The Required Manifest section displays the PCRs used for attestation. The **Show Manifest** button displays the manifest data (i.e. Digest Value, Event Name, etc.) for the Component stored in the MLE.

Note: Only the Description field can be modified.



Whitelist > Edit MLE > Add MLE

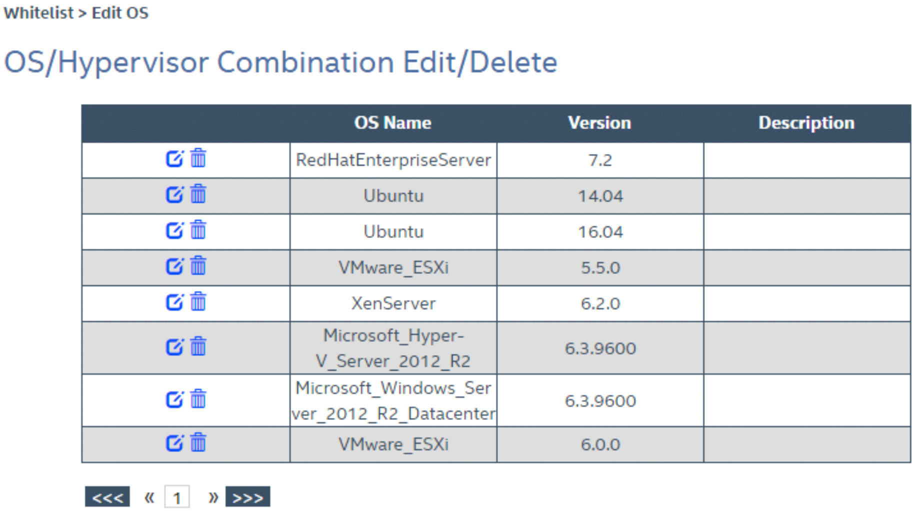
New Measured Launch Environment (MLE) Configuration

MLE Type:	BIOS
Host OS:	Intel Corporation
VMM Name:	Intel_Corporation
VMM Version:	01.00.0060
Attestation Type:	PCR
Description:	
Whitelist Host:	
Required Manifests:	☑0 □1 □2 □3 □4 □5 □12 □13 □14 ☑17 □18 □19 □20
Manifest List :	Show Manifest

4.2.7 Edit OS

Select **Whitelist > Edit OS**.

OS records are created automatically when MLEs are imported.


Whitelist > Edit OS

OS/Hypervisor Combination Edit/Delete

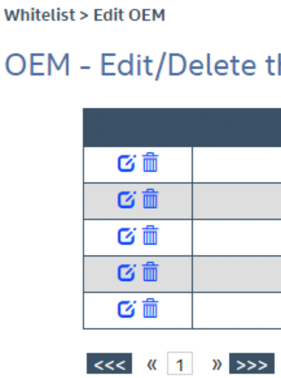
	OS Name	Version	Description
✎ 🗑	RedHatEnterpriseServer	7.2	
✎ 🗑	Ubuntu	14.04	
✎ 🗑	Ubuntu	16.04	
✎ 🗑	VMware_ESXi	5.5.0	
✎ 🗑	XenServer	6.2.0	
✎ 🗑	Microsoft_Hyper-V_Server_2012_R2	6.3.9600	
✎ 🗑	Microsoft_Windows_Server_2012_R2_Datacenter	6.3.9600	
✎ 🗑	VMware_ESXi	6.0.0	

Using the **Edit** icon to edit an OS only allows the **Description** field to be populated or changed. An OS can only be deleted using the trash can icon if no MLEs are currently configured to use the OS.

4.2.8 Edit OEM

Select **Whitelist > Edit OEM**.

OEM records are created automatically when MLEs are imported.



Whitelist > Edit OEM

OEM - Edit/Delete the Values

	Name	Description
✎🗑	Cisco Systems, Inc.	Cisco Systems, Inc.
✎🗑	Intel Corporation	Intel Corporation
✎🗑	Intel Corp.	Intel Corp.
✎🗑	Dell Inc.	Dell Inc.
✎🗑	American Megatrends Inc.	American Megatrends Inc.

Editing an OEM only enables the **Description** field to be populated or changed. An OEM can only be deleted if no MLEs are currently configured to use the OS.

4.3 Host Management

4.3.1 Importing Hosts

Host registration can be performed using a flat (text) file, or by importing hosts from a VMWare vCenter cluster.

To use the flat file method, provide a plain text file (*.txt* format) containing a list of hosts to be registered in the following format(s), either IP Address or hostname can be used.

For Open Source and Windows hosts:

intel|myTrustAgent|https://MyTrustAgent:1443

microsoft|mywindowshost|https://<*hostname*>:1443

For VMWare hosts:

vmware|*Host_Name*|*Vmware_Connection_String*[1]

For example:

vmware|myVmwareHost|https://192.168.1.1:443/sdk;*Username*;*Password*.

For Citrix Xen:

citrix|*Host_Name*|*Citrix_Connection_String*[1]

For example:

citrix|myCitrixHost|https://<*hostname*>:443/;*Username*;*Password*.

Specific examples for both flat file and vCenter cluster registrations are described in the sections that follow.

[1] The connection string must be a valid URL. Note that the / after the port number is required.

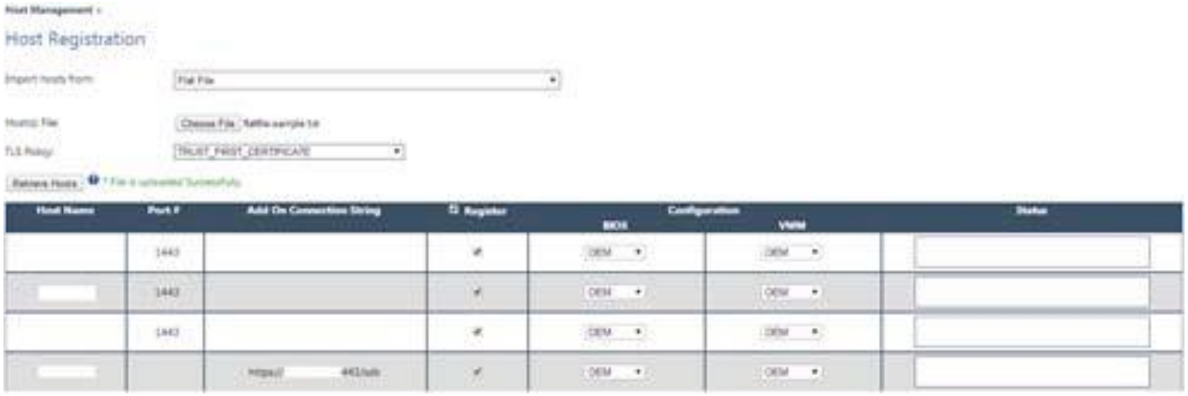
In each case, each host to be registered has drop-down selections for the BIOS and VMM MLEs to be used. These selections correspond to the MLEs that have been created using the whitelist tools. Note that only created MLE types are selectable (for example, if only an OEM MLE has been created, the Global selection does not appear in the list). Intel recommends using the OEM MLE for both BIOS and VMM selections unless a specific business requirement indicates otherwise.

An appropriate BIOS and VMM MLE must exist for each host prior to registration. For instance, to register a host that has a BIOS version X, a BIOS MLE for BIOS version X must already exist. If appropriate MLEs do not exist, an error message is generated indicating that the missing VMM/OS/BIOS/version information was not found.

A shared TLS policy must be selected for use with all hosts listed. To register hosts with host-specific TLS policies, the individual host registration function (**Host Management > Add Host**) must be used. For additional details on TLS policies, refer to Section 5.5, "TLS Policy Overview".

4.3.2 Registering Hosts Using a Flat File

1. Select **Host Management > Import...**



Host Management >

Host Registration

Import hosts from: Flat File

Host(s) File: Choose File flatfile-sample.txt

TLS Policy: TRUST_FIRST_CERTIFICATE

Retrieve Hosts * File is uploaded Successfully

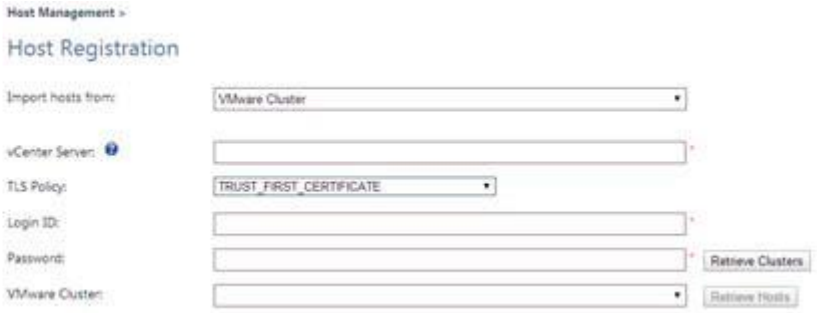
Host Name	Port #	Add On Connection String	☐ Register	Configuration		Status
				BIOS	VMM	
	1443		✓	OEM	OEM	
[redacted]	1443		✓	OEM	OEM	
	1443		✓	OEM	OEM	
[redacted]		https:// [redacted] 443/sdk	✓	OEM	OEM	

2. Open a text file and enter the host information in the format previously described.
3. The **Flat File** option is selected by default. Click **Browse** and then go to the location of the text file created in the previous step.
4. Select a shared TLS policy for use with all hosts listed in the flat file. For more information on TLS policies, refer to the Section 4.3.11.5, "TLS Policy Management".
5. Click **Retrieve Hosts** to import the list of hosts and their connection details from the flat file. By default all hosts are selected.
6. Click **Register Host** to register the selected hosts.

   Note: For each host, there is a drop-down selection for the BIOS and VMM MLEs to be used for this specific host (OEM, Global, or Host). The selected MLE is used for attesting the host. Make the selection most appropriate for your data center. Generally, Intel recommends using the OEM MLE type under most circumstances for both BIOS and VMM MLEs. Details on whitelist MLE management can be found in Section 5.4.

4.3.3 Registering ESXi Hosts by Cluster

1. Select **Host Management > Import...**

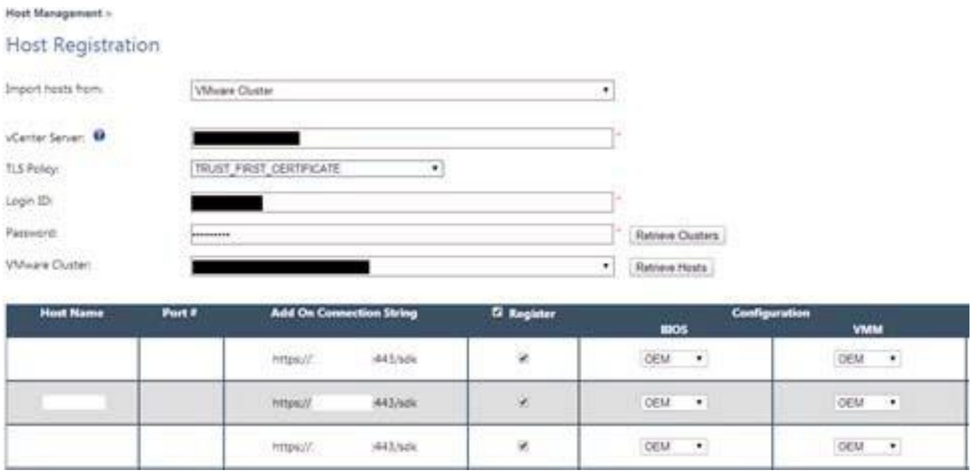


Host Management >

Host Registration

Import hosts from:	VMware Cluster ▾
vCenter Server:	*
TLS Policy:	TRUST_FIRST_CERTIFICATE ▾
Login ID:	*
Password:	*
VMware Cluster:	▾

2. Select **VMware Cluster** from the drop-down list.
3. Enter the IP Address or DNS name of the vCenter server and login credentials for an account with administrator rights in vCenter server.
4. Select a shared TLS policy for use with this vCenter server. Note that all ESXi hosts share the same TLS policy used by the vCenter server they are managed by. For more information on TLS policies, refer to the Section 4.3.11.5, "TLS Policy Management".
5. Click **Retrieve Clusters**.
6. A list of all clusters in vCenter become available in the **VMware Cluster** drop-down list. Select the cluster you want to import hosts from and click **Retrieve Hosts** to import the list of hosts in that cluster from vCenter Server. By default all hosts are selected. Click **Register Host** to register the selected hosts.



Host Management >

Host Registration

Import hosts from:	VMware Cluster ▾
vCenter Server:	[redacted] *
TLS Policy:	TRUST_FIRST_CERTIFICATE ▾
Login ID:	[redacted] *
Password:	•••••••••• *
VMware Cluster:	[redacted] ▾

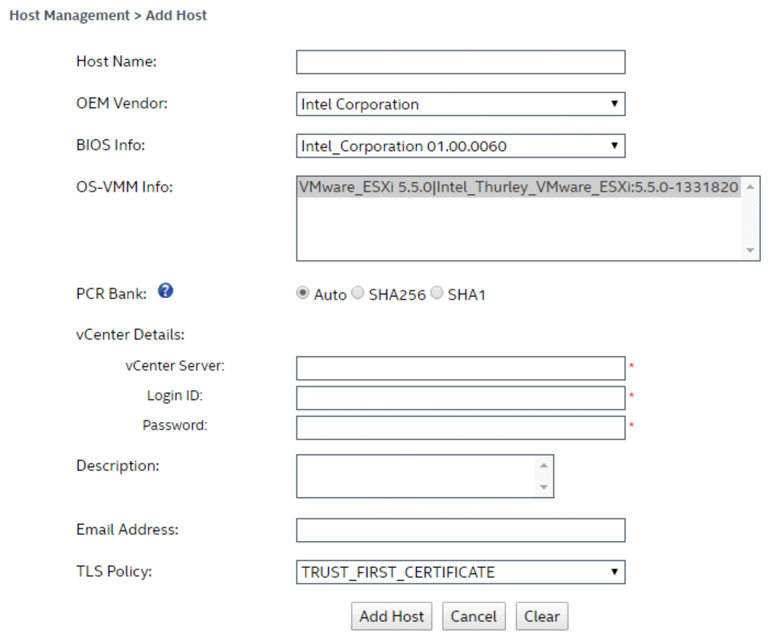
Host Name	Port #	Add On Connection String	☑ Register	Configuration	
				BIOS	VMM
		https:// :443/sdk	✓	OEM ▾	OEM ▾
		https:// :443/sdk	✓	OEM ▾	OEM ▾
		https:// :443/sdk	✓	OEM ▾	OEM ▾

Note: For each host, there is a drop-down selection for the BIOS and VMM MLEs to be used for this specific host. Only MLEs that have already been created appear in the drop-down list. For, example, if a global MLE exists but no OEM MLEs exist, the drop-down list display only a Global option). The selected MLE is used for attesting the host. Make the selection most appropriate for your data center. Generally, Intel recommends using the OEM MLE type under most circumstances for both BIOS and VMM MLEs). Details on Whitelist MLE management can be found in Section 5.4.

4.3.4 Manual Host Registration

The **Add Host** tool enables hosts to be registered individually, and enables more detailed control over the MLEs used for attestation. Because the **Import** tool automatically detects the BIOS and OS-VMM versions, the manual registration tool enables any BIOS and VMM MLE (within limits, such as attempting to register a vCenter host using a Trust Agent VMM MLE does not work because the vCenter connection details are missing) to be used to attest the host. Be aware that selecting a mismatched MLE results in the host appearing as untrusted when attested. Intel generally advises using the automated **Import** tool for host registration.

1. Select **Host Management > Add Host**.



Host Management > Add Host

Host Name:	
OEM Vendor:	Intel Corporation
BIOS Info:	Intel_Corporation 01.00.0060
OS-VMM Info:	VMware_ESXi 5.5.0|Intel_Thurley_VMware_ESXi:5.5.0-1331820
PCR Bank:	● Auto ○ SHA256 ○ SHA1
vCenter Details:	
vCenter Server:	*
Login ID:	*
Password:	*
Description:	
Email Address:	
TLS Policy:	TRUST_FIRST_CERTIFICATE

2. Enter the hostname or IP Address for the host to be registered.
3. Select the **OEM Vendor** from the drop-down list.
4. Select the appropriate BIOS version from the drop-down list in the **BIOS Info** field.

   Note: This is the BIOS MLE and is populated according to the OEM vendor selected.

5. Select the OS-VMM combination from the list in the **OS-VMM Info** field. This is the VMM MLE.
6. Select the PCR Bank. It is recommended to use the **Auto** selection.

   Note: This option only applies to a host with TPM 2.0 and is ignored for a host with TPM 1.2. Since TPM 2.0 may support multiple PCR banks, this option allows the user to choose a specific PCR bank for attestation with a TPM 2.0 host.

- For Auto-Default selection, the system chooses the strongest algorithm in the PCR banks that the host TPM supports.
- For SHA256 selection, SHA256 PCR bank for a TPM 2.0 host is selected.
- For SHA1 selection, SHA1 PCR bank for a TPM 2.0 host is selected.

7. Enter the connection details.
    - For vCenter, enter the hostname or IP Address of the vCenter server that manages the host, and credentials for a vCenter user with administrative rights.
    - For Citrix Xen, enter the port number (Citrix Xen uses port 443 by default) and the administrative credentials for the host.
    - For Trust Agent hosts, enter the port number for the Trust Agent and Windows (default 1443).
8. Enter a Description (optional).
9. Enter an email address (optional).
10. Select a TLS policy for use with this host (either a new host-specific policy or an existing shared policy. For more information on TLS policies, refer to the Section 4.3.11.5, "TLS Policy Management").
11. Click **Add Host** to register the host.

4.3.5 Trust Dashboard

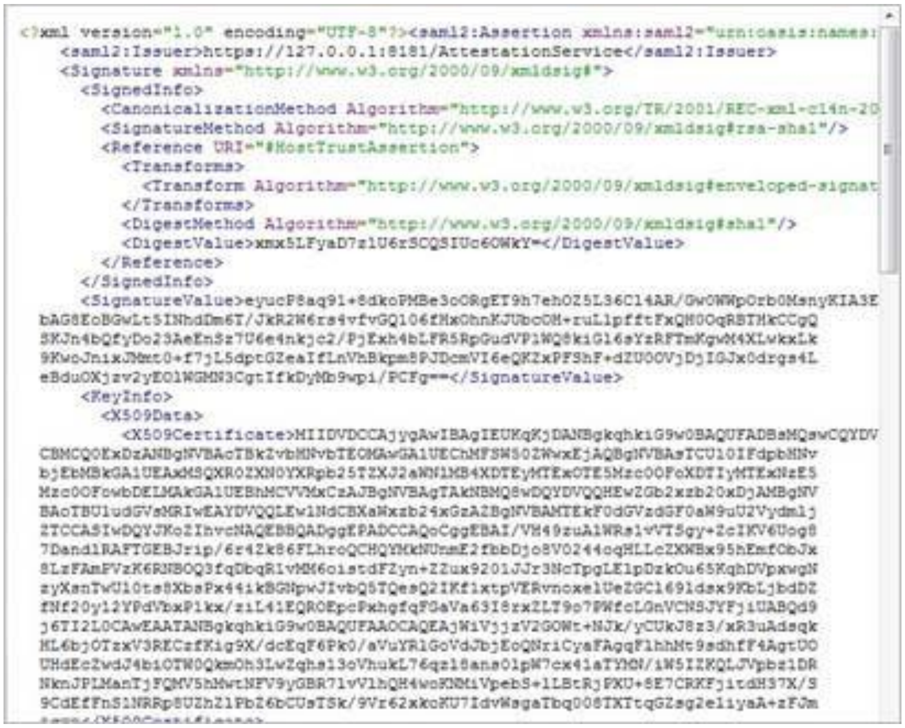
Select **Trust > Dashboard**.

The Trust Dashboard displays and refreshes the trust attestation of all hosts registered to Intel® CIT.

- Trusted hosts appear with a green icons.
- Un-trusted hosts appear as red icons.
- Hosts with no trust status (hosts with TXT disabled or PCR/Module information otherwise unavailable) appear with a blue question mark icon.
- If no Asset Tag Certificates have been created for a host, the Asset Tag Status icon appears as a blue question mark icon; after provisioning, the icon appears as green or red according to whether the Asset Tag information matches the Asset Certificate. Refer to the *Asset Tag Provisioning Guide* for details on Asset Tag functionality.
- To refresh the trust attestation of any host, click Refresh in the **Trust Status** field of the specific host.

4.3.6 Trust Assertion Details

The **Trust Assertion Details** tool displays the Security Assertion Markup Language (SAML) assertion in *.xml* format.

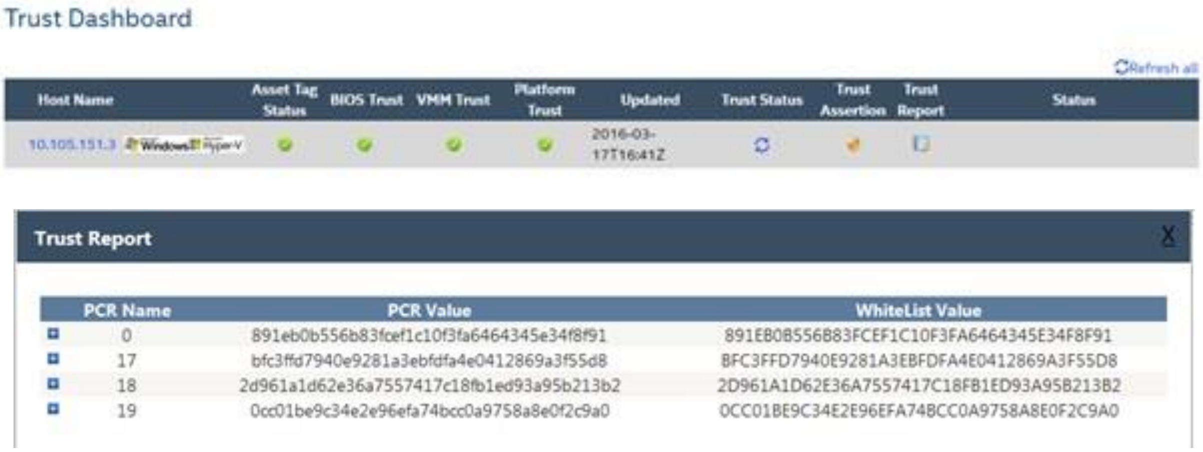


```
<?xml version="1.0" encoding="UTF-8"?><saml2:Assertion xmlns:saml2="urn:oasis:names:
   <saml2:Issuer>https://127.0.0.1:8181/AttestationService</saml2:Issuer>
   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
     <SignedInfo>
       <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20
       <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
       <Reference URI="#HostTrustAssertion">
         <Transforms>
           <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signat
         </Transforms>
         <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <DigestValue>xmx5LFyaD7z1U6rSCQSIUc6OWkY=</DigestValue>
       </Reference>
     </SignedInfo>
     <SignatureValue>eyucP8aq91+8dkoPMBe3oORgET9h7ehOZ5L36C14AR/GwOWWpOrb0ManyKIA3E
 bAG8EoBGwLt5INhdDm6T/JkR2W6rs4vfvGQ106fMxOhnKJUbcOM+ruLlpfftFxQH0OqRBTMkCCgQ
 SKJn4bQfyDo23AeEnSz7U6e4nkjc2/PjExh4bLFR5RpGudVP1WQ8kiG16sYzRFTmKgwM4XLwkxLk
 9KwoJnixJMmt0+f7jL5dptGZeaIfLnVhBkpm8PJDcmVI6eQKZxPFShF+dZU0OVjDjIGJx0drgs4L
 eBduOXjzv2yEOlWGMN3CgtIfkDyMb9wpi/PCFg==</SignatureValue>
     <KeyInfo>
       <X509Data>
         <X509Certificate>MIIDVDCCAjygAwIBAgIEUKqKjDANBgkqhkiG9w0BAQUFADBsMQswCQYDV
 CBMCQ0ExDzANBgNVBAcTBkZvbHNvbTEOMAwGA1UEChMFSW50ZWwxEjAQBgNVBAsTCU10IFdpbHNv
 bjEbMBkGA1UEAxMSQXR0ZXN0YXRpb25TZXJ2aWN1MB4XDTEyMTExOTE5Mzc0OFoXDTIyMTExNzE5
 Mzc0OFowbDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAkNBMQ8wDQYDVQQHEwZGb2xzb20xDjAMBgNV
 BAoTBU1udGVsMRIwEAYDVQQLEw1NdCBXaWxzb24xGzAZBgNVBAMTEkF0dGVzdGF0aW9uU2Vydmlj
 ZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAI/VH49zuA1WRs1vVT5gy+ZcIKV6Uog8
 7Dand1RAFTGEBJrip/6r4Zk86FLhroQCHQYMkNUnmE2fbbDjo8V0244oqHLLcZXWBx95hEmfObJx
 8LzFAmFVzK6RNBOQ3fqDbqR1vMM6oistdFZyn+ZZux9201JJr3NcTpgLE1pDzkOu65KqhDVpxwgN
 zyXsnTwU10ts8XbsPx44ikBGNpwJIvbQ5TQesQ2IKf1xtpVERvnoxelUeZGC169ldsx9KbLjbdDZ
 fNf20y12YPdVbxP1kx/ziL41EQROEpcPxhgfqFGaVa63I8rxZLT9o7PWfcLGnVCN8JYFjiUABQd9
 j6TI2L0CAwEAATANBgkqhkiG9w0BAQUFAAOCAQEAjWiVjjzV2GOWt+NJk/yCUkJ8z3/xR3uAdsqk
 HL6bjOTzxV3RECzfKig9X/dcEqF6Pk0/aVuYR1GoVdJbjEoQNriCyaFAggF1hhMt9sdhfF4AgtUO
 UHdEcZwdJ4biOTW0QkmOh3LwZqhs13oVhukL76qz18ans01pW7cx41aTYMN/iW5IZKQLJVpbz1DR
 NknJPLManTjFQMV5hMwtNFV9yGBR71vV1hQH4woKNMiVpebS+1LBtRjPXU+8E7CRKFjitdH37X/S
 9CdEfFnS1NRRp8UZhZ1PbZ6bCUsTSk/9Vr62xKoKU7IdvWsgaTbq008TXTtqGZsg2eliyaA+zFJm
 a===</X509Certificate>
```

4.3.7 Trust Report

The **Trust Report** tool displays the PCR values for the selected host, as well as the whitelisted MLE values for that host's BIOS and VMM versions.

Note: Different PCRs will be reported for different operating systems.



Trust Dashboard

↻Refresh all

Host Name	Asset Tag Status	BIOS Trust	VMM Trust	Platform Trust	Updated	Trust Status	Trust Assertion	Trust Report	Status
10.105.151.3 Windows Hyper-V	✓	✓	✓	✓	2016-03-17T16:41Z	↻	✎	🗎	

Trust Report X

	PCR Name	PCR Value	WhiteList Value
⊞	0	891eb0b556b83fcef1c10f3fa6464345e34f8f91	891EB0B556B83FCEF1C10F3FA6464345E34F8F91
⊞	17	bfc3ffd7940e9281a3ebfdfa4e0412869a3f55d8	BFC3FFD7940E9281A3EBFDFA4E0412869A3F55D8
⊞	18	2d961a1d62e36a7557417c18fb1ed93a95b213b2	2D961A1D62E36A7557417C18FB1ED93A95B213B2
⊞	19	0cc01be9c34e2e96efa74bcc0a9758a8e0f2c9a0	0CC01BE9C34E2E96EFA74BCC0A9758A8E0F2C9A0

4.3.8 Asset Tag Provisioning

There are two methods supported for provisioning asset tags to assets. The push method involves creating an asset certificate using the asset's UUID and a tag selection and pushing the certificate from the asset tag service to the asset. The pull method involves booting the asset itself to a provisioning image, which requests a new asset certificate from the asset tag service using a given selection.

Note: Currently, the push method is only supported for Citrix XenServer with the XenServer Measured Boot supplemental pack installed, and Trust Agent hosts. The pull method is supported for Citrix Xen, VMWare ESXi, and KVM.

4.3.8.1 Push Provisioning Method (Manual Certificate Creation)

The push method for asset tag provisioning is initiated by the asset tag management service. The administrator creates a new asset certificate and provisions the certificate to a provisioning agent running on the asset. This section describes the steps needed to create a new asset certificate and provision it to a host using the push method.

Note: Because the asset must have a provisioning agent running to write the asset certificate to the TPM, only Citrix Xen hosts with the XenServer Measured Boot supplemental pack installed, or open-source Xen or KVM hosts with the trust agent installed can currently be provisioned using the push method. ESXi hosts must be provisioned using the pull method.

Prerequisites:

- One or more Citrix Xen or trust agent hosts registered in Intel® Cloud Integrity Technology.
- Appropriate asset tags and asset selections to provision to the host(s) must have been created.

Procedure:

1. From the Intel® CIT portal, browse to **Asset Tag Management** > **Certificate Management**.
2. Type the IP or hostname (as it appears in Intel® CIT), or the hardware UUID of the server to be provisioned in the **Hostname** field.
3. Using the **Tag Selection** drop-down box, choose the tag selection to be provisioned to the host, and click **Save**.

   This creates a new asset certificate, and imports it into the Intel® CIT database. The certificate is matched to the appropriate host for attestation using the host's hardware UUID.
4. Under **Search Certificates**, leaving all fields blank, click **Search** (this lists all asset certificates in the database). Users should see a certificate listed with the UUID entered.
5. To the right of the certificate details, are several hyper-links. Click **Deploy**.
6. New fields appear below the certificate list. Enter the IP address or hostname of the host (along with the username and password for an account with root privileges if the host is a Citrix XenServer). Click **Provision**. The asset certificate is provisioned host's TPM.
7. Refresh the host status on the Intel® CIT server Trust Dashboard. The Asset Tag icon should turn green.

   Note: For Citrix hosts, reboot to complete the provisioning process. Certificate information is extended to PCR 22 after the reboot.

4.3.8.2 Push Provisioning Method (Automated)

The push method for asset tag provisioning is initiated by the asset tag management service. The administrator creates a new asset certificate and provisions the certificate to a provisioning agent running on the asset. This section describes the steps needed to automatically generate and provision asset certificates for one or more hosts concurrently using identical tags.

Note: Because the asset must have a provisioning agent running in order to write the asset certificate to the TPM, only Citrix Xen hosts with the XenServer measured boot supplemental pack installed, or open-source Xen or KVM hosts with the trust agent installed can currently be provisioned using the push method. ESXi hosts must be provisioned using the pull method.

Prerequisites:

- One or more Citrix Xen or trust agent hosts registered in Intel® Cloud Integrity Technology.
- Appropriate asset tags and asset selections to provision to the host(s) must have been created.

Procedure:

1. From the Intel® CIT portal, browse to **Asset Tag Management** > **Provision Tags**.
2. Select the servers to be provisioned from the **Available Servers** list by double-clicking each server to be provisioned.
3. In the **Tag Selection** box, choose either a pre-existing selection from the **Available Selections** tab, or select **Upload XML** to use an XML file.
4. Click the **Provision** button at the bottom of the page. This automatically generates new asset certificates for each host using the specified selection of tags, and provision the asset certificates to all servers in the Servers to Provision list.
5. Refresh the trust status on the Intel® CIT server Trust Dashboard.

4.3.8.3 Pull Provisioning Method

The pull method for asset tag provisioning is initiated from the host by the provisioning agent. The agent works in two ways: it can send the UUID of the host to receive a pre-generated valid certificate if one exists, or the agent can send an XML file to the asset tag service, which creates certificates with tag selections according to the options in XML. In either case, the SHA1 hash value of the resulting certificate is written to the host's TPM.

The asset tag provisioning agent must be run from an operating system with certain prerequisites installed, and therefore is provided on an ISO-format disk image (assettag.iso). This image is used with iPXE on a provisioning network to boot to the provisioning image remotely.

4.3.8.4 PXE Image Configuration (Pull Provisioning)

Prerequisites:

- A working iPXE server with supporting technologies and an NFS share.
- A deployed instance of the asset tag management service.
- The asset tag provisioning image (*assettag.iso*).
- The TPM for any hosts to be provisioned using the pull method must be in the clear state with Intel® TXT activated.

Note: If provisioning a host running the trust agent using the pull method and the host is already registered for attestation in Intel® CIT, the TPM ownership does NOT need to be cleared. For trust agent hosts, boot to PXE, allow the script to run, and then boot back into the normal OS.

Procedure:

1. Extract the contents of *assettag.iso* and copy the *casper* folder to the Network File System (NFS) share directory on the PXE server.
2. Copy the Secure Sockets Layer (SSL) certificate from your asset tag management server (located in */etc/intel/cloudsecurity/ssl.crt.pem*) to the PXE server and place it in the NFS share directory of the PXE server.
3. On the PXE server, edit */tftpboot/ipxe/bootloader.cfg* and add the following arguments:

```
atag_cert='http://<PXE IP Address>/<nfsshare>/ssl.crt.pem'
atag_username='admin'
atag_password='password' xml='<Path_To_XML>'
atag_server='http://<IP Address>:<Port>/mtwilson/v2'
```

Note: Replace *<PXE IP Address>* with the IP address or hostname of the PXE server. Replace *<nfsshare>* with the path to the NFS share. Replace *<IP Address>* and *<Port>* with the IP address or hostname and port of the asset tag management server on the PXE network. Refer to the following sample *bootloader.cfg* file:

```
#!ipxe bootloader.cfg

cpuid --ext 29 && set arch ia32e || set arch ia32
echo !!!!PXEBooting to Assettag Provisioning Agent..!!!!
sleep 2
set nfs-root=<PXE_Server_IP>:/var/www/nfsshare
kernel nfs://<PXE_Server_IP>/var/www/nfsshare/casper/vmlinuz
initrd nfs://<PXE_Server_IP>/var/www/nfsshare/casper/initrd.gz
echo got the initrd and vmlinuz kernel
sleep 2
echo now booting...
sleep 1
echo .....
imgargs vmlinuz root=/dev/nfs boot=casper netboot=nfs nfsroot=<PXE_server_IP>:/var/ www/
nfsshare atag_username='admin' atag_password='password' atag_cert='http://
<PXE_server_IP>/nfsshare/ ssl.crt.pem ' atag_xml='<path_to_xml>' atag_server='http://
<Asset Tag Server IP>:<Asset Tag Server Port>/mtwilson/v2'
boot
```

4.3.8.5 Asset Tag Provisioning Agent Script (PXE)

Prerequisites:

- An iPXE server configured.
- The TPM of any ESXi hosts or trust agent hosts that are not registered in Intel® CIT must be set to the clear state and Intel® TXT/TPM must be active.

To provision a host, reboot it and use the **Boot Options** menu to boot to the provisioning PXE network during startup. The asset tag provisioning agent script automatically runs when the image loads. The script contacts the Intel® CIT server and retrieve the latest valid asset certificate for the host's hardware UUID. If no certificate exists, new certificates can be created using an XML file containing host UUIDs and the tag name/value pairs that should be assigned to them.

Procedure:

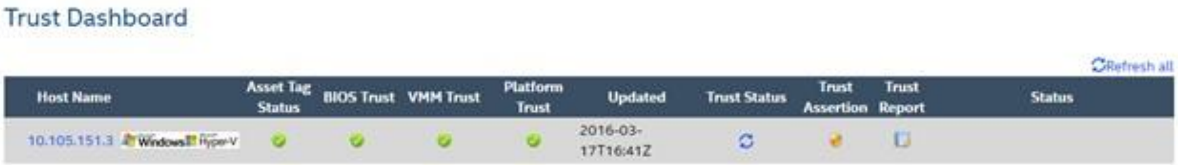
1. Reboot the host to be provisioned.
2. Enter the boot menu and select the network device on the PXE network. The system boots and the provisioning agent script runs.
3. If the host is running ESXi or Citrix XenServer or is not yet registered in Intel® CIT, enter the BIOS and re-enable TPM (the script automatically clears the TPM ownership).

   Note: This step is not required for trust agent hosts that are registered in Intel® CIT.
4. Reboot the host to its normal operating system.

4.3.8.6 Asset Tag Visibility and Attestation

After tag selections have been provisioned to hosts, the Intel® CIT Attestation Server can provide visibility and attestation for the asset tags. Registering a host with the Attestation Server after the host has been provisioned with asset tags enables the Attestation Server Trust Dashboard to display the tags provisioned to the host, and enables Intel® CIT to attest to the validity of the Asset Certificate.

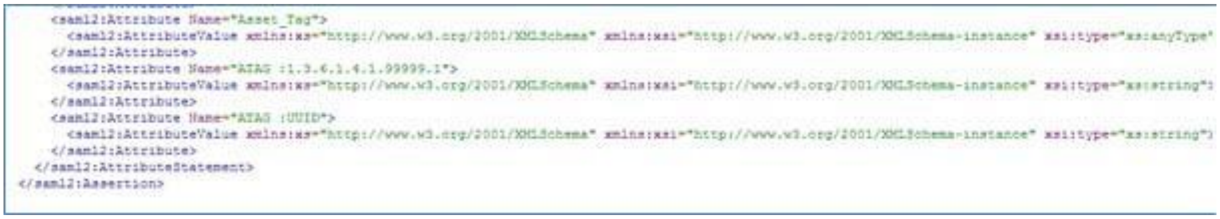
After provisioning completes, follow the standard procedures for whitelisting and host registration. After hosts are registered, the Attestation Server Trust Dashboard page displays the new Asset Tag trust status. Hovering with the mouse over the Asset Tag trust icon displays the tags provisioned to the host.



Trust Dashboard

Host Name	Asset Tag Status	BIOS Trust	VMM Trust	Platform Trust	Updated	Trust Status	Trust Assertion	Trust Report	Status
10.105.151.3 Windows Hyper-V	●	●	●	●	2016-03-17T16:41Z	⟳	●	□	

The Trust Assertion now also contains the Asset Tag information, including the name/value pairs associated with the host.

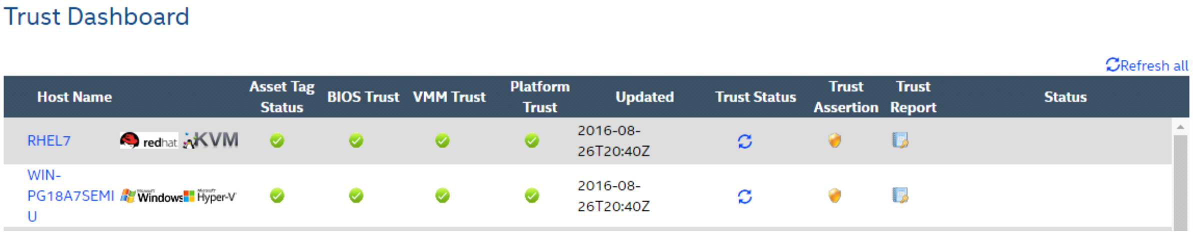


```
    <saml2:Attribute Name="Asset_Tag">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:anyType"
    </saml2:Attribute>
    <saml2:Attribute Name="ATAG :1.3.6.1.4.1.99999.1">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
    </saml2:Attribute>
    <saml2:Attribute Name="ATAG :UUID">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
    </saml2:Attribute>
  </saml2:AttributeStatement>
</saml2:Assertion>
```

4.3.9 Bulk Trust Refresh

Select **Trust > Refresh All**.

The **Refresh all** icon enables the trust attestation of multiple hosts to be refreshed concurrently (as opposed to individually clicking **Refresh** for each host on the Trust Dashboard). By default, clicking the **Refresh all** icon will update the Trust Status of all registered hosts.



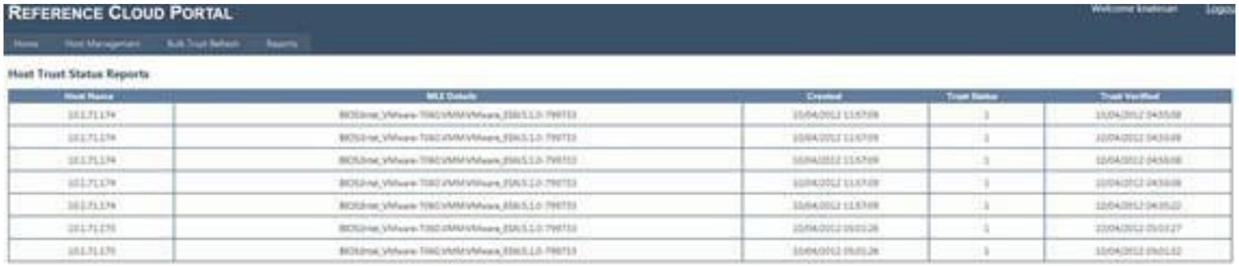
Trust Dashboard

Host Name		Asset Tag Status	BIOS Trust	VMM Trust	Platform Trust	Updated	Trust Status	Trust Assertion	Trust Report	Status
RHEL7	redhat KVM	✓	✓	✓	✓	2016-08-26T20:40Z				
WIN-PG18A7SEMIU	Windows Hyper-V	✓	✓	✓	✓	2016-08-26T20:40Z				

4.3.10 Reports

Select **Trust > Report**.

This tool enables the retrieval of reports detailing the creation date, MLE information, trust status, and the date of the last trust status refresh. Select the hosts to be reported and then click **Get Report**.



Host Trust Status Reports

Host Name	MLE Details	Created	Trust Status	Trust Verified
10.1.71.174	BIOS:Intel_VMware-T060:VMM:VMware_ESXi:5.1.0-799733	10/04/2012 11:57:09	1	10/04/2012 04:55:58
10.1.71.174	BIOS:Intel_VMware-T060:VMM:VMware_ESXi:5.1.0-799733	10/04/2012 11:57:09	1	10/04/2012 04:50:49
10.1.71.174	BIOS:Intel_VMware-T060:VMM:VMware_ESXi:5.1.0-799733	10/04/2012 11:57:09	1	10/04/2012 04:55:08
10.1.71.174	BIOS:Intel_VMware-T060:VMM:VMware_ESXi:5.1.0-799733	10/04/2012 11:57:09	1	10/04/2012 04:50:09
10.1.71.174	BIOS:Intel_VMware-T060:VMM:VMware_ESXi:5.1.0-799733	10/04/2012 11:57:09	1	10/04/2012 04:35:22
10.1.71.175	BIOS:Intel_VMware-T060:VMM:VMware_ESXi:5.1.0-799733	10/04/2012 05:01:26	1	10/04/2012 05:03:27
10.1.71.175	BIOS:Intel_VMware-T060:VMM:VMware_ESXi:5.1.0-799733	10/04/2012 05:01:26	1	10/04/2012 05:01:32

4.3.11 Administration

4.3.11.1 User Account Registration

While the default user (**admin** by default, unless renamed during setup) has access to all Intel® CIT Attestation Server roles and can perform all tasks, the Attestation Server supports the ability to add new users and to limit the tasks they can perform. Intel recommends that, after the Attestation Server is deployed, one or more new user accounts are created and that access to the administrative account is restricted.

Intel® Cloud Integrity Technology 2.2 adds the ability to create new roles. Roles are defined as a collection of one or more permissions that can then collectively be applied to users for access control. While the included Attestation Server *User Interface* (UI) does not incorporate functionality for making new roles, these functions are available programmatically (refer to Javadocs for full REST API documentation).

Currently, permissions consist of a domain, an action, and a selection.

- Domains represent groups of objects. For example, hosts or MLEs would define permissions relating to host registration and MLE creation.
- Actions represent the specific permissions allowed within the domain. For instance, one permission might define access to the create action of the hosts domain, meaning this permission would enable the creation of new host objects (such as registering hosts with the Attestation Server).
- Selections represent a filter that can be used to narrow down which specific items within a domain the permission applies to. Currently, an asterisk (*) can be used to refer to "all selections".

To maintain backwards compatibility, the roles from Intel® Cloud Integrity Technology 1.x have been re-created in the new security framework, and are included in the initial installation.

A description of the various roles and the procedure for creating new users follows.

4.3.11.2 Description of Intel® Cloud Integrity Technology Roles

Following are the Intel® CIT roles:

- **Security** — Manage access to Intel® CIT APIs, approve/reject access requests, and assign roles.
- **Whitelist** — Add/edit/delete OEM, OS, MLE records and import white-list MLEs.
- **Attestation** — Add/edit/delete host records, assign MLE to a host, and get host trust status.
- **Report** — Get host trust status.
- **Audit** — Manage audit records.
- **AssetTagManagement** — Create/delete/provision Asset Tags.

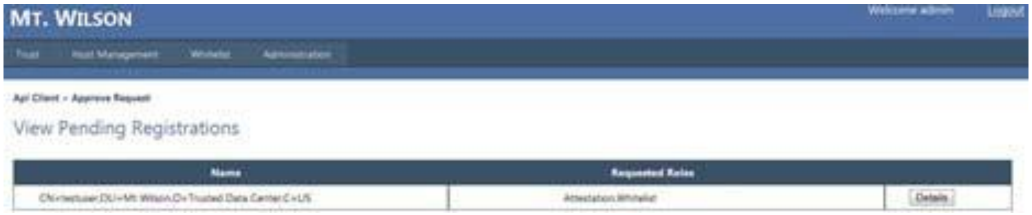
4.3.11.3 New User Creation

To create a new user:

1. Connect to the Intel® CIT web portal.

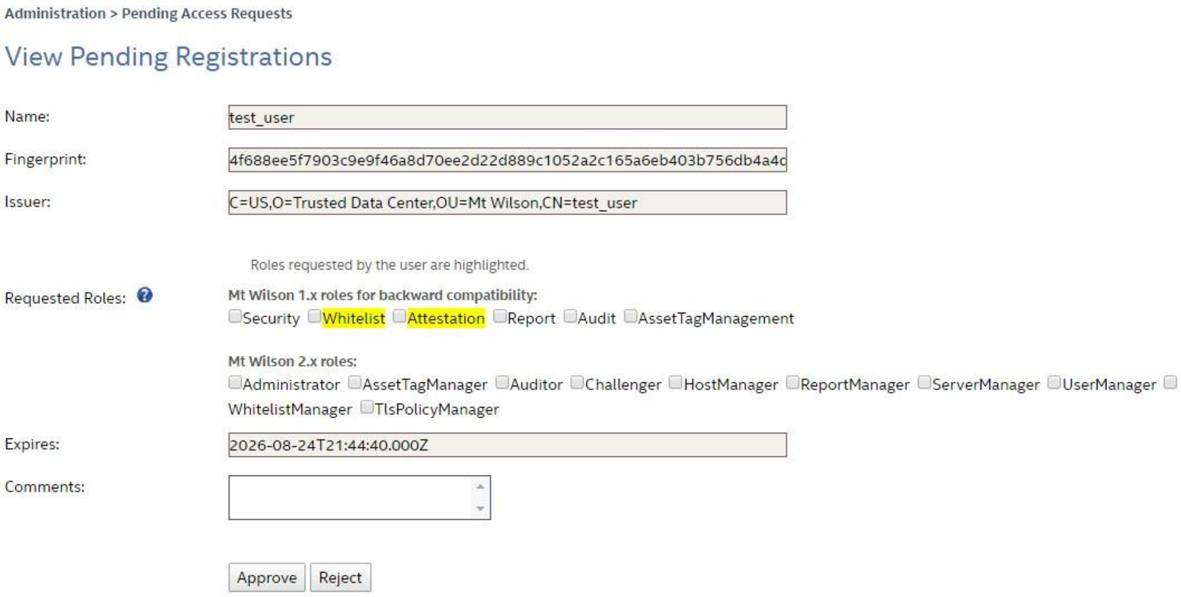
   https://<*Attestation_Server_IP_Address*>:<*port*>/mtwilson-portal/index.html

2. Click **Register** and create a new user.
   a. Enter the username.
   b. Select a locale.
   c. Enter the password.
   d. Confirm the password.
3. Login to the Intel® CIT Attestation Server portal with admin credentials (the values of MC_FIRST_USERNAME and MC_FIRST_PASSWORD in the *mtwilson.env* file that was used for installation).
4. Select **Administration > Pending Requests** from the menu bar.
5. The portal displays a list of pending user account approvals. To approve the registered user, click **Details**.



Name	Requested Roles	
CN=testuser,OU=Mt Wilson,O=Trusted Data Center,C=US	Attestation,Whitelist	Details

6. Select the required roles for the new user, enter any comments, then click **Approve**.



Administration > Pending Access Requests

View Pending Registrations

Name:	test_user
Fingerprint:	4f688ee5f7903c9e9f46a8d70ee2d22d889c1052a2c165a6eb403b756db4a4c
Issuer:	C=US,O=Trusted Data Center,OU=Mt Wilson,CN=test_user

Roles requested by the user are highlighted.

Requested Roles:

Mt Wilson 1.x roles for backward compatibility:

□Security □Whitelist □Attestation □Report □Audit □AssetTagManagement

Mt Wilson 2.x roles:

□Administrator □AssetTagManager □Auditor □Challenger □HostManager □ReportManager □ServerManager □UserManager □ WhitelistManager □TlsPolicyManager

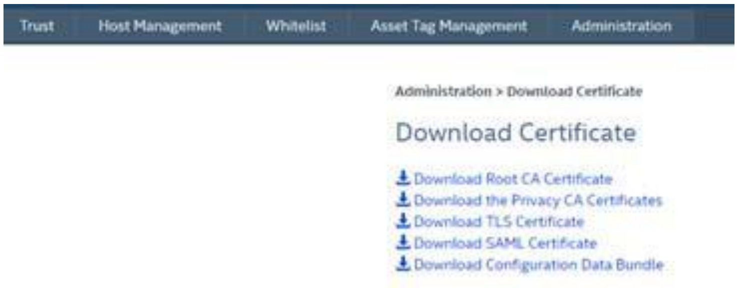
Expires:	2026-08-24T21:44:40.000Z
Comments:	

4.3.11.4 View Certificates

Select **Administration > View Certificates**.

The **View Certificates** tool enables the various certificates used by the Intel® CIT Attestation Server (including the Root CA certificate, the PrivacyCA certificate, the TLS certificate, and the SAML certificate) to be downloaded.

Intel recommends backing up these certificates after installation. The default location for these certificates is the */etc/intel/cloudsecurity* folder.



Administration > Download Certificate

Download Certificate

- Download Root CA Certificate
- Download the Privacy CA Certificates
- Download TLS Certificate
- Download SAML Certificate
- Download Configuration Data Bundle

4.3.11.5 TLS Policy Management

Select **Administration > TLS Policy Management**.

The **TLS Policy Management** page enables user to create, modify, and delete shared-scope TLS policies. A TLS policy represents an authorized TLS certificate to which the Attestation Server connects. Different TLS Policy types include raw certificate and certificate hash models, and can be applied to individual hosts or shared across many hosts. Information from the TLS certificate(s) of any host(s) to which the Attestation Server connects needs to be added using the TLS Policy Management interface. The specific TLS policy to be used for a given host connection can be selected during host registration.

A TLS policy can be created for each individual host, or, if all TLS certificates across a large number of hosts use the same root certificate authority, the root certificate can be used in the Attestation Server TLS Policy. This policy could then be used for all server connections using the same root certificate.

For ESXi hosts, the TLS certificate is managed by vCenter Server. For any given number of ESXi hosts to be registered in the Attestation Server, only the TLS certificate of the vCenter Server that manages those ESXi hosts needs to be added to an Attestation Server TLS Policy.

For legacy compatibility, the Intel® CIT 1.x TLS policies TRUST_FIRST_CERTIFICATE and INSECURE can still be used, but are not enabled by default. To enable one or both of these TLS policies, set the MTW_TLS_POLICY_ALLOW variable in *mtwilson.env* prior to installation. For example:

```
export MTW_TLS_POLICY_ALLOW=certificate,certificate-digest,public-key,public-
keydigest,TRUST_FIRST_CERTIFICATE,INSECURE
```

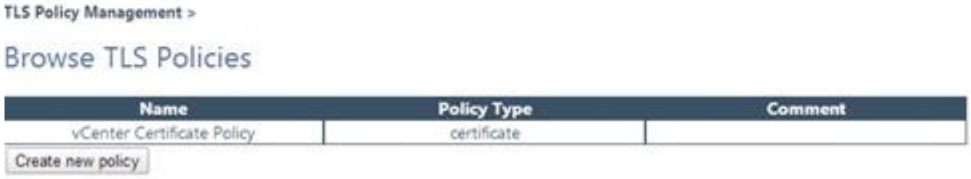
To set the default TLS policy, use the MTW_DEFAULT_TLS_POLICY_ID variable. For example:

```
export MTW_DEFAULT_TLS_POLICY_ID=”TRUST_FIRST_CERTIFICATE".
```

The TRUST_FIRST_CERTIFICATE TLS policy records the first TLS certificate used by a remote server and accepts connections to that server using only that TLS certificate. This option requires no management by the administrator, and does not need a specific policy to be created once it is enabled.

The INSECURE TLS Policy effectively ignores TLS certificate verification.

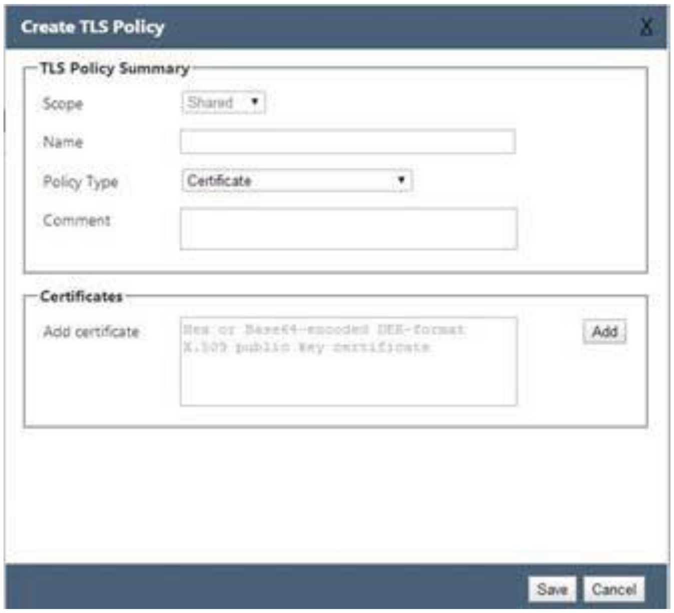
Note: Per-host TLS policies are created at the time of whitelisting or registering individual hosts. To edit per-host TLS policies, use the **Edit Host** function for the appropriate host. Per-host TLS policies are deleted when the host they are attached to is itself deleted.



Browse TLS Policies

Name	Policy Type	Comment
vCenter Certificate Policy	certificate	

1. To edit or delete an existing policy, click the name and select the appropriate choice from the pop-up menu.
2. To create a new shared TLS policy, click **Create New Policy**.



Create TLS Policy

TLS Policy Summary

Scope: Shared

Name

Policy Type: Certificate

Comment

Certificates

Add certificate: Hex or Base64-encoded DER-format X.509 public key certificate

3. Enter a name for the new policy in the **Name** field.
4. Select the **Policy Type**.
   - For **Certificate** policies, the entire TLS certificate from the remote server needs to be copied (Hex or Base-64 encoded DER-format X.509 public key certificate).
   - For **Certificate Fingerprint**, copy the SHA-1, SHA-256, or SHA-384 hash of the remote server TLS certificate.
   - For **Public Key**, copy the entire Hex or Base64-encoded DER-format X.509 public key.
   - For **Public Key Fingerprint**, copy the SHA-1, SHA-256, or SHA-384 hash of the remote server public key.
5. Add a comment in the **Comment** field (optional).
6. Add the certificate, public key, or hash as appropriate for the TLS policy type.

   Note: The **Add Certificate** field provides a reminder of what is required and in what format.
7. Click **Save** to save the policy. The new policy now is available for selection to assign to hosts.

5.0 Configuration

5.1 PCR Definitions

PCR	Measurement Parameters	Description	Operating System
PCR 0	BIOS ROM and Flash Image	This PCR is based solely on the BIOS version, and remains identical across all hosts using the same BIOS. This PCR is used as the BIOS MLE (PCRs 1-5 can also be used in a BIOS MLE, but this is not recommended).	• All
PCR 12	Data events and highly volatile events	This PCR measures some of the modules which has boot counters in it. It changes on every boot and resume (Microsoft Windows ONLY)	• Microsoft Windows Server 2012 • Microsoft Hyper-V 2012
PCR 13	Boot Module Details	This PCR remains static except major changes such as kernel module update, different device driver for different OEM servers, etc. (Microsoft Windows ONLY)	• Microsoft Windows Server 2012 • Microsoft Hyper-V 2012
PCR 14	Boot Authorities	Used to record the Public keys of authorities that sign OS components. Expected not to change often. (Microsoft Windows ONLY)	• Microsoft Windows Server 2012 • Microsoft Hyper-V 2012
PCR 17	ACM	This PCR measures the SINIT ACM, and is platform-specific. This PCR is part of the BIOS MLE. TPM1.2 measurements are not modular. TPM 2.0 measures the SINIT ACM and platform specific modules for Linux hosts.	• VMware ESXi • Citrix Xen • Ubuntu • Red Hat Enterprise Linux
PCR 18	MLE [Tboot +VMM]	This PCR measures the tboot and hypervisor version. In ESXi hosts, only the tboot version is measured.	• VMware ESXi • Citrix Xen • Ubuntu • Red Hat Enterprise Linux
PCR 19	OS Specific. • ESXi — non Kernel modules • Citrix Xen and TA hosts — OS + Init RD + UUID	For ESXi, this PCR contains individual measurements of all of the non-Kernel modules. For Citrix Xen and TA hosts, this PCR is a measurement of the OS, InitRD, and UUID. This changes with every install due to InitRD and UUID change.	• VMware ESXi • Citrix Xen • Ubuntu • Red Hat Enterprise Linux
PCR 20	For ESXi only. VM Kernel and VMK Boot	This PCR is used only by ESXi hosts and is blank for all other host types.	• VMware ESXi
PCR 22	Asset Tag	This PCR contains the measurement of the SHA1 of the Asset Tag Certificate provisioned to the TPM, if any.	• VMware ESXi • Citrix Xen

5.2 Tested Platforms

OEM	Model
DELL*	R720
	R710
CISCO* UCS	B200M3
	C220M3
	C240M3
	5108
HP*	BL460C G8
	DL380 G7
Intel® EPSD	S2600GZ
	S5520UR
	S2600WT2
	S2600WT2R
IBM*	X3650M4
Super Micro*	X9DRD-iF
	X10SLH-F/X10SLM+-F
Quanta*	D51B-1U

5.3 Whitelisting Guidelines

Whitelisting consists of extracting known-good MLE values to use for later attestation. While a known-good host is used to provide these values, the MLEs are stored in the Attestation Service database and are only changed when deleted or overwritten by user action. If the known-good host changes for any reason, the MLEs already captured are not updated unless a user with the appropriate roles repeats the white-listing procedure and overwrites the MLEs with new, updated values. Registration of the known-good host for attestation is optional.

MLEs are divided into two types: BIOS and VMM. To register a given host, MLEs appropriate for that host's BIOS and OS/hypervisor configuration must already exist in the Intel® CIT database.

For BIOS MLEs:

- Select the **OEM** option to provide whitelist for each OEM type and BIOS version. All the hosts of a given OEM type & BIOS version are verified with this whitelist.
- Select the **HOST** option to provide whitelist by specific host and BIOS version.

For VMM/OS MLEs:

- Select the **OEM** option to provide whitelist for each OEM type and VMM/OS version. All the hosts of a given OEM type platform, & VMM/OS version are verified with this whitelist.
- Select the **HOST** option to provide whitelist by host and VMM/OS version. This whitelist is used to verify ONLY the specified host.
- Select the **GLOBAL** option to provide the whitelist for each VMM/OS version and build number. All servers/hosts across all OEM types using the same platform run this specified VMM/OS version and build are verified with this whitelist.

Note: OEM and global VMM MLEs are specific to server platforms. If your data center consists of a single OS/hypervisor version but has dual-platform based servers, a separate VMM MLE is required for each of the two platforms even though the OS/hypervisor configuration is otherwise uniform.

5.4 MLE Administration

Intel® CIT automatically generates MLE names based on their classification as VMM or BIOLS MLEs, the specific type of MLE (Global, OEM, or Host), and details captured from the host during the import. These values include the server platform (Romley, Thurley, etc), the name and version numbers of the OEM, OS, BIOS, and hypervisor as appropriate. For example,

IBM_Thurley_VMware_ESXi_5.1.0-799733

is an OEM-level VMM MLE imported from an IBM Thurley server running ESXi version 5.1.0 build 799733. This VMM MLE can be used to attest the OS/hypervisor component of any IBM Thurley-based server running ESXi 5.1.0 build 799733, but cannot be used to attest an HP or Dell host, or any host running a different version of ESXi or a different hypervisor.

Typically, only a single MLE is needed for a given platform and hypervisor or BIOS version. However, there are some use cases where different modules are present in a VMM MLE even though the version remains the same. For example, using the Cisco Nexus 1000 virtual switch adds several module entries for a VMware ESXi host, and yet the version and build number for ESXi does not change. If only some hosts in your data center are using the Cisco Nexus 1000, you would need two separate VMM MLEs to attest those hosts with and without the additional modules. Otherwise, one set of hosts attest as untrusted depending on whether the host used to import the MLE was configured to use the virtual switch. Hosts attest as untrusted if:

- They contain one or more modules that do not exist in the whitelist MLE.
- They do not contain one or more modules that do exist in the whitelist MLE.
- Any hash values for PCRs or modules for the host do not match those values in the MLE.

Important note for ESXi hosts and VMware High Availability:

When High Availability is enabled for a cluster of ESXi hosts, vCenter pushes an agent to each host in the cluster. This agent is included in the modules attested by the Attestation Server, but is not measured until the next reboot of each affected host. Intel recommends rebooting each host in such a cluster after High Availability is enabled so that the VMM measurements are uniform.

Important Note for Microsoft Windows Server 2012:

Modifying the installed features (Active Directory service, DNS, Hyper-V, and so on) on a Microsoft Windows Server system causes the extended measurement of PCR 13 and/or PCR 14 to change. Intel recommends rebooting hosts after enabling/disabling/upgrading features to ensure that the current measurements in the TPM correspond to the actual system configuration, so that the VMM measurements are uniform.

When importing a new MLE when an existing MLE of the same version information already exists in the database, Intel® CIT automatically checks to see whether any PCRs or modules have changed. If they have, the default behavior is to add a new MLE without overwriting the original, with a numerical tag at the end of the name (001, 002, and so on). This can be overridden in the UI by selecting the **Overwrite** option on the **Import Whitelist MLE** page. This causes the Intel® CIT to overwrite the existing MLE regardless of whether any PCRs or modules have changed. Only the MLE of the selected type (VMM or BIOS, global, OEM, or host) and identical versions are overwritten.

In the case of an environment upgrade (like upgrading from one ESXi version to another), it is important to import the new whitelist MLE value first, before upgrading the remainder of the environment. After importing the new, updated MLEs, perform the production upgrade and then re-register the hosts. By re-registering the hosts, they are automatically assigned to the correct, updated MLEs.

5.5 TLS Policy Overview

Intel® CIT validates the authenticity of connections through the use of various TLS verification policies.

5.5.1 TLS Policy Types

The Intel® CIT Attestation Service uses six types of TLS policies:

Policy Type	Behavior	Shared	Per-Host
Certificate	The certificate policy requires one or more trusted certificates or CA certificates and only connects to a peer whose certificate either is a trusted certificate or is signed by a CA that is trusted. This policy type also performs hostname verification. ***Note:*** The remote server's hostname must be resolvable from the Intel® CIT server).	Yes	Yes
Certificate Fingerprint	This policy stores the SHA1 hash of the certificate for validation rather than the entire certificate itself.	Yes	Yes
Public Key	The public key policy requires one public key parameter and only connects to a peer using that key. This is similar to SSH public key authentication of clients and hosts. Hostname verification is NOT performed when using public key TLS policies.	Yes	Yes
Public Key Fingerprint	This policy stores the SHA1 hash of the public key for validation rather than the public key itself.	Yes	Yes
TRUST_FIRST_CERTIFICATE	This policy stores the first certificate encountered when connecting to a host, and uses that certificate for all future TLS validation with that host. This was the default TLS policy used in Intel® Cloud Integrity Technology 1.x, and is disabled by default.	Yes	No
INSECURE	This policy disables all TLS validation. All connections are accepted regardless of TLS certificates. This is disabled by default.	Yes	No

5.5.2 Policy Scope

TLS policies can be per-host or shared across multiple hosts.

- **Per-Host** — A per-host TLS policy is an individual, per-host TLS policy. When the host is deleted, its per-host TLS policy is automatically deleted as well. Policies using the per-host scope can be created on the individual host registration and whitelisting UI pages, and edited on the **Edit Host** page for the appropriate host.
- **Shared** — A shared TLS policy might be referenced by multiple host records. When a host is deleted that referenced a shared TLS policy, the shared policy continues to exist regardless if there are any remaining hosts that are referencing it. Shared policies must be explicitly deleted by the user. Shared policies of all types can be created and edited under the **Administration** tab.

Intel® CIT requires a TLS policy to be defined for any remote host to which it connects. If no TLS policy is defined, or if the TLS information does not match the TLS policy, the connection fails.

5.5.3 Default Policy Selection

Any shared-scope policy can be defined as the "default" TLS policy for a given Intel® CIT environment. For example, if all TLS certificates for all hosts in the attestation environment have been signed by the same CA certificate, that CA certificate can be used to create a shared-scope certificate policy, and this same policy could be used to validate all TLS connections with all attested hosts. By configuring this policy as the default TLS policy, Intel® CIT uses this specific policy for all hosts unless another policy is specified.

In the Intel® CIT UI, this mostly means that the default policy is automatically selected from the drop-down list when registering hosts. From an API perspective, it means that, when calling a registration API, if no TLS policy is specifically defined in the call, the default TLS policy is used. Using a shared default policy that is valid across all hosts in the attestation environment can greatly simplify TLS policy and host management.

Note: During installation, the only two shared-scope policies that might be available are TRUST_FIRST_CERTIFICATE and INSECURE, and these only if they have actually been enabled. All other policies must be user-created after installation.

To define a default TLS policy, edit the *mtwilson.properties* file and set the value of *mtwilson.default.tls.policy.id* to either the UUID or the name of the shared-scope TLS policy to be set as the default. Restart Intel® CIT to affect the change.

5.5.4 Intel® Cloud Integrity Technology 1.x Behavior

In Intel® Cloud Integrity Technology 1.x, only two policies were available to validate the TLS certificates of remote hosts: TRUST_FIRST_CERTIFICATE (default) and INSECURE.

- **TRUST_FIRST_CERTIFICATE** recorded the first TLS certificate encountered when connecting to a host, and used that certificate to validate future connections.
- **INSECURE** turned off all TLS certificate validation entirely (all connections were trusted regardless of TLS certificates).

To configure Intel® Cloud Integrity Technology 2.2 to use the same TLS policy behavior as was used in Intel® Cloud Integrity Technology 1.x, configure the *mtwilson.properties* file with the following settings:

```
mtwilson.tls.policy.allow=TRUST_FIRST_CERTIFICATE
mtwilson.default.tls.policy.id=TRUST_FIRST_CERTIFICATE
```

This can be done automatically during installation by setting the following variables in *mtwilson.env*:

```
export MTW_TLS_POLICY_ALLOW= TRUST_FIRST_CERTIFICATE
export MTW_DEFAULT_TLS_POLICY_ID=TRUST_FIRST_CERTIFICATE
```

5.6 Database Configuration for Remote Database Servers

By default, Intel® Cloud Integrity Technology deploys its own database on a local machine, it is also possible to use a remote database server. If using an external server is required, the remote database must be configured to accept remote connections.

For Postgres, edit the */etc/postgresql/9.3/main/pg_hba.conf* file and add the following line:

```
host     all     all     <IP RANGE>/<SUBNET>     password
```

Using a remote Postgres database also requires modifying the *postgresql.conf* file (in the same folder as *pg_hba.conf*) to include the Intel® CIT server in the *listen_addresses* setting, which is commented out by default.

```
# - Connection Settings -

listen_addresses = '*'
```

Setting this value to “'*'” enables connections from all addresses. This can be restricted to only the Intel® CIT server(s), or to a specific IP range that includes the Intel® CIT server(s).

Prior to running the Intel® CIT installation, the Intel® CIT user (*root* by default) and the database schema (*mw_as* by default) must be created on the remote database server, and full rights over the database must be given to the user. These steps are not required if the database resides on the same machine as the Intel® CIT services.

5.7 SSL Changes from Intel® CIT 1.x to Intel® CIT 2.x

Intel® Cloud Integrity Technology 1.x releases use a default policy of TRUST_FIRST_CERTIFICATE that assumes that hosts are in a secure network when being registered with Intel® CIT. If there is an attacker on the network during registration, it is possible to mount a man-in-the-middle attack by exploiting this policy. It is possible to change the default policy, but other policies have limited support in Intel® CIT 1.x.

Intel® Cloud Integrity Technology 2.2 releases improve on this by enabling the client to specify the TLS policy for each host registration as either insecure, a known public key, or a trusted certificate with hostname verification (with public key or certificates provided during registration). However, this change is only available in the v2 APIs because it would break compatibility with 1.x clients if it were made available in the v1 APIs.

Therefore, it is important to note that hosts registered using the v1 APIs of Intel® Cloud Integrity Technology 2.2 releases continue to use the server-configured default policy. When additional security is required, hosts should be registered using the v2 APIs that enable the TLS policy to be specified by the client.

For additional details, refer to Section 4.3.11.5, “TLS Policy Management”.

5.8 Command-Line Interface

5.8.1 Attestation Service

This section describes commands that can be run from the command line on the Attestation Service server.

5.8.1.1 Check Server Version

The following command displays the server version:

```
mtwilson version
MtWilson Linux Utility
Version 3.0-SNAPSHOT
Build <date> (<source branch>)
```

5.8.1.2 Check Server Status

To check the server status:

```
mtwilson status
Checking Tomcat process... Running (pid <pid>)
Checking if mtwilson is deployed on webserver...Deployed
Checking if mtwilson-portal is deployed on webserver... Deployed
```

5.8.1.3 Start and Stop the Server

To start the Attestation Server:

```
mtwilson start
```

To stop the Attestation Server:

```
mtwilson stop
```

To restart the Attestation Server:

```
mtwilson restart
```

5.8.1.4 Change the Database Password and Update Configuration Files

This change the current database password and updates the relevant Attestation Service configuration files with the new password.

```
mtwilson change-db-pass
```

5.8.1.5 Output Attestation Service SSH Key and SAML Certificate Fingerprints

```
mtwilson fingerprint

== SSH HOST KEYS ==
98:07:72:61:42:54:86:96:11:c0:07:9b:ac:8d:c8:67 (ECDSA-256)
98:de:31:42:80:e0:61:ae:db:ab:9d:df:09:e3:45:f8 (DSA-1024)
22:95:2a:72:e2:41:94:f2:08:20:0e:76:fd:39:00:da (RSA-2048)
88:bf:60:02:05:d5:26:62:03:79:09:e7:96:26:b5:83 (ED25519-256)
== MT WILSON SAML CERTIFICATE ==
Owner: CN=mtwilson, OU=Mt Wilson, O=Intel, L=Folsom, ST=CA, C=US
Valid from: Fri Aug 21 13:01:28 PDT 2015 until: Mon Aug 18 13:01:28 PDT 2025
         MD5:  36:9D:B2:D2:12:6B:12:4C:9D:51:C5:00:38:0D:31:F4
         SHA1: 07:88:E6:41:32:CE:D9:6E:A0:94:0E:17:D2:2F:4E:89:1C:32:51:98
```

5.8.1.6 Detect and Output Currently-Installed Version of and Installation Location Java

This command is used for troubleshooting. If the Attestation Service cannot detect a valid version of Java, installation and functionality fail. The Java version must be equal or greater than the specified minimum Java version in the installation answer file.

```
mtwilson java-detect
JAVA_HOME=/usr/share/jdk1.7.0_51
java_bindir=
java=/usr/share/jdk1.7.0_51/bin/java
```

5.8.1.7 Detect and Output Currently Installed Version and installation location of Tomcat

```
mtwilson tomcat-detect
TOMCAT_HOME=/opt/mtwilson/share/apache-tomcat-7.0.34
tomcat_bin=/opt/mtwilson/share/apache-tomcat-7.0.34/bin/catalina.sh
tomcat="env
PATH=/opt/mtwilson/share/jdk1.7.0_51/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/
usr/bin:/sbin:/bin:
JAVA_HOME=/opt/mtwilson/share/jdk1.7.0_51
CATALINA_HOME=/opt/mtwilson/share/apache-tomcat-7.0.34
CATALINA_BASE=/opt/mtwilson/share/apache-tomcat 7.0.34
CATALINA_CONF=/opt/mtwilson/share/apache-tomcat-7.0.34/conf/opt/mtwilson/share/
apache-tomcat-7.0.34/bin/catalina.sh"
```

5.8.1.8 Generate a New Tomcat SSL Certificate

```
mtwilson tomcat-sslcert
```

5.8.1.9 Check the Status of the Tomcat Web Server

This command also outputs the Tomcat TLS certificate fingerprint information.

```
mtwilson tomcat-status
Checking Tomcat process... Running (pid 31418)
Owner: CN=10.1.71.130, CN=127.0.0.1, OU=Mt Wilson, O=Trusted Data Center, C=US
Valid from: Fri Aug 21 13:01:18 PDT 2015 until: Mon Aug 18 13:01:18 PDT 2025
         MD5:  DF:AE:44:60:12:10:5D:2B:04:CE:BE:0A:AE:D6:45:39
         SHA1: C5:B8:5D:90:E1:3D:42:58:F0:13:02:38:54:C9:66:4C:0D:C3:27:AF
```

5.8.1.10 Backup and Restore All Keys, Certificates, and Secrets Used by the Attestation Service

These commands backup and restore all Attestation Service keys, certificates, and secrets to/from an encrypted blob file. These can be used to back up and restore keys when performing Attestation Service re-installations, and to restore functionality and access to data in the event of an accidental deletion.

An encryption password must be set before running these commands. This password is used to encrypt or decrypt the secrets to/from the encrypted blob file. To set the encryption password, export the *MTWILSON_PASSWORD* variable with a password value.

```
export MTWILSON_PASSWORD=<password>

mtwilson key-backup
Keys backed up to: /var/mtwilson/key-backup/mtwilson-keys_2015-09-03.135329.enc

mtwilson key-restore <path_to_file>
mtwilson key-restore /var/mtwilson/key-backup/mtwilson-keys_2015-09-03.135329.enc
Keys restored from: /var/mtwilson/key-backup/mtwilson-keys_2015-09-03.135329.enc
```

5.8.1.11 Execute All Setup Tasks

The setup tasks are documented in the next section. The following command executes all setup tasks, in order:

```
mtwilson setup
```

Any environment variables needed by the setup tasks should be set before running the command. When all the setup tasks complete successfully, there is no output on the screen. Errors and warnings are shown on the screen.

5.8.1.12 Execute Specific Setup Tasks

If you need to execute a specific setup task, you can name one or more tasks to execute in order on the command line. The following example executes two tasks:

```
mtwilson setup task1 task2
```

5.8.1.13 Validate a Setup Task Without Executing It

If you need to check that a setup task has completed successfully, but do not want to execute it (validation or dry run), you can use the **--noexec** option.

```
mtwilson setup <tasks...> --no-exec
```

5.8.1.14 Force Execution of a Setup Task Even if It is Already Validated

Normally, if a setup task has already been successfully completed, its execution is skipped. If you want to force that task to run again, you can use the **--force** option.

```
mtwilson setup <tasks...> --force
```

5.8.1.15 Continue Executing Subsequent Setup Tasks Even if One Fails

Normally, if a setup task fails to complete, subsequent tasks are skipped. If you want to try running subsequent tasks anyway, you can use the **--continue** option.

```
mtwilson setup <tasks...> --continue
```

5.8.1.16 Uninstall

The uninstall command removes the Attestation Server files but keeps the configuration, keys, and any credentials for remote services. To completely remove the Attestation Service and its data, use the **zeroize**, **erase-data**, and **erase-users -all** commands first, because as they are not available after uninstall.

```
mtwilson uninstall
```

To erase all data (excluding user information) from the Attestation Service database:

```
mtwilson erase-data
```

To erase user data from the Attestation Service database:

```
mtwilson erase-users < --all >
```

By default, **erase-users** deletes all users except those with the word “admin” in their names. To delete all users including administrative users, use the **-all** switch. Note that this terminates all access to the Attestation Service, but the data remains in the database.

To immediately delete all certificates and cryptographic keys from the Attestation Service and the database:

```
mtwilson zeroize
```

5.8.1.17 Help

Displays basic command options.

```
mtwilson help
Usage: mtwilson {change-db-pass|erase-data|erase-users|fingerprint|help|
                 glassfish-detect|glassfish-enable-logging|glassfish-sslcert|
                 glassfish-status|java-detect|mysql-detect|mysql-sslcert|
                 tomcat-detect|tomcat-sslcert|tomcat-status|key-backup|
                 key-restore|restart|setup|start|status|stop|uninstall|version|
                 zeroize}
```

5.8.2 Trust Agent

The following commands can be run from the command line for both Linux and Microsoft Windows hosts.

5.8.2.1 Start

```
tagent start
Started trust agent
```

5.8.2.2 Stop

```
tagent stop
Stopped trust agent
```

5.8.2.3 Restart

```
tagent restart
Stopped trust agent
Started trust agent
```

5.8.2.4 Status

```
tagent status
Trust agent is running
```

Or, if the service is not running, this displays instead:

```
Trust agent is not running
```

5.8.2.5 Version

```
tagent version
trustagent Version 2.0-SNAPSHOT
Build release-2.0 at 2014-07-16T15:37:28.702-0700
```

5.8.2.6 Authorize

This command forces the Trust Agent to re-establish all certificates and authentication with the Attestation Server. This can be used if, for example, the Attestation Server has been re-installed and now uses new certificates.

```
tagent authorize
Stopped trust agent
Started trust agent
```

5.8.2.7 Setup

This command is fully described in the Trust Agent subsection of the Installation and Configuration Options section of this guide.

```
tagent setup --stdout
```

5.8.2.8 Uninstall

This command uninstalls the Trust Agent, but does not remove any other component. If the Trust Agent is uninstalled and then re-installed, the host record in the Attestation Service is not automatically removed. After re-installing the Trust Agent, the host should be re-registered in the Attestation Server.

```
tagent uninstall --purge
Stopped trust agent
Stopping Monit service...
```

5.8.2.9 Help

```
tagent help
Usage: /usr/local/bin/tagent start|stop|authorize|start-http-server|version
Usage: /usr/local/bin/tagent setup [--force|--noexec] [task1 task2 ...]
Available setup tasks:
configure-from-environment
create-keystore-password
create-tls-keypair
create-admin-user
create-tpm-owner-secret
create-tpm-srk-secret
create-aik-secret
take-ownership
download-mtwilson-tls-certificate
download-mtwilson-privacy-ca-certificate
request-endorsement-certificate
request-aik-certificate
register-tpm-password
```

5.9 Installation and Configuration Options

5.9.1 Attestation Service

The Attestation Service can be configured during installation using an answer file. If an answer file is not provided or a required option is missing, the installer prompts the user for the required information. The answer file must be named "*mtwilson.env*" and should be placed in the same directory as the installer, or in the */root/* directory.

After installation, configuration is managed by several *.properties* files located in the */etc/intel/cloudsecurity/* directory. These files are encrypted by default. Use caution when modifying these files, as incorrect configuration can cause the Attestation Service to stop functioning.

5.9.1.1 Installation Options

Table 1. Installation Options (set in *mtwilson.env*)

Environment Variable	Default Value or Example	Notes
ADD_POSTGRESQL_REPO	yes	Required to download and install PostgreSQL 9.3 if installing on Ubuntu 12.04. Unnecessary if PostgreSQL 9.3 has already been downloaded.
DATABASE_HOSTNAME	127.0.0.1	The IP Address or hostname of the database server. This can be the local machine (default) if the database resides on the Attestation Service VM, or can be set to a remote server's IP Address or hostname if the database is set up elsewhere. See the configuration instructions for remote databases in Section 5.6.
DATABASE_PASSWORD *mtwilson.db.password*	mypassword	The password for the Attestation Service database.
DATABASE_PORTNUM	5432	The port number of the database server. By default, use 5432 for PostgreSQL.
DATABASE_SCHEMA	mw_as	The schema name for the Attestation Service database. The default is "mw_as". If a schema name other than the default is desired, it must be manually created on the database server, along with a user with full privileges over the schema, before the Attestation Service installation is performed. The installer only creates a database schema named "mw_as", but can use a pre-existing schema of any name
DATABASE_USERNAME *mtwilson.db.user*	root	The user name for the Attestation Service database.
DATABASE_VENDOR	postgres	The database vendor. Currently the only accepted value is **postgres**.
INSTALL_LOG_FILE	*/tmp/mtwilson-install.log*	Sets the path for the installation log.

Table 1. Installation Options (set in *mtwilson.env*) (Continued)

Environment Variable	Default Value or Example	Notes
INSTALL_PKGS	"postgres java tomcat logrotate privacyca SERVICES PORTALS"	Tells the installer which packages to be installed as a space-separated list. It takes the following possible options: • **postgres** (Installs the postgresql database server) • **java** (Installs Java 7u51 SDK) • **tomcat** (Installs the tomcat web server), • **privacyca** (Installs the Attestation Service certificate authority) • **SERVICES** (Installs the non-portal Attestation Service services) • **PORTALS** (Installs the Attestation Service UI portal) • **monit** (Installs the monit process monitor for process availability) • **logrotate** (installs the logrotate log management service)
INSTALLED_MARKER_FILE	*/var/opt/intel/.mtwilsonInstalled*	Creates an installation flag in the designated location that can be used by ISVs to confirm whether the Attestation Service installation has been completed.
JAVA_REQUIRED_VERSION	1.7.0_51	The minimum required Java version. The value shown is the default.
LOG_COMPRESS	compress	Compresses old rotated logs in *.zip* format. This setting used is only if **Logrotate** is installed.
LOG_COPYTRUNCATE	copytruncate	Truncate the original log file in place after creating a copy, instead of moving the old log file and optionally creating a new one. This setting used is only if **Logrotate** is installed.
LOG_DELAYCOMPRESS	delaycompress	Postpones compression of older rotated log files until the next log rotation. This setting used is only if **Logrotate** is installed.
LOG_OLD	2	Sets the number of log rotations to retain. This setting used is only if **Logrotate** is installed.
LOG_ROTATION_PERIOD	daily	Sets the time period used for log rotation. Acceptable values can include daily, hourly, weekly, monthly, and yearly. This setting used is only if **Logrotate** is installed.
LOG_SIZE	100M	Creates a new log file and rotates at this size threshold. This setting used is only if **Logrotate** is installed.

Table 1. Installation Options (set in *mtwilson.env*) (Continued)

Environment Variable	Default Value or Example	Notes
MC_FIRST_PASSWORD	mypassword	The password for the administrative user defined in *MC_FIRST_USERNAME*.
MC_FIRST_USERNAME	admin	The username that is used for the initial default administrative user. This user account is used to approve any additional users created through the **Registration** link on the logon page of the portal.
MTWILSON_API_BASEURL	https://${MTWILSON_SERVER}:8443/mtwilson/v1	The Base URL for Intel® CIT v1 APIs. This value should be set to: *"https://${MTWILSON_SERVER}:<PORT>/mtwilson/v1"* Replace *<PORT>* with the port for the web server that is used (8443 for Tomcat by default).
MTWILSON_OWNER	root	Sets the user that owns the Intel® CIT installation (*root* by default).This user is created if it does not already exist.
MTWILSON_OWNER_PASSWORD	mypasword	Sets the password for the Intel® CIT owner user specified above if it does not already exist. Not used if the user already exists (does not change existing passwords).
MTWILSON_SERVER	*Myserver.mydomain.com*	The IP Address or hostname of the Attestation Service server.
MTWILSON_TAG_ADMIN_PASSWORD	mypassword	Password for the account that is used for pull tag provisioning.
MTWILSON_TAG_ADMIN_USERNAME	tagadmin	Username for the account that is used for pull tag provisioning.
MTWILSON_TAG_CERT_IMPORT_AUTO	TRUE	If set to **TRUE** (default), all asset tag certificates are automatically imported to be matched to an appropriate host for tag attestation. If set to **FALSE**, the certificates are not imported automatically. Attestation of asset tag information depends on the certificate being imported, so it is recommended that this be set to **TRUE** for most deployments.
MTWILSON_TAG_HTML5_DIR	*/usr/share/apache-tomcat-7.0.34/webapps/mtwilson-portal/tag*	The path to the asset tag HTML5 information. This path can change if the web service has been installed in a directory other than the path used by default.
MTWILSON_TAG_KEY_PASS	mypassword	Intel® CIT tag key password.
MTWILSON_TAG_KEYSTORE	*/opt/mtwilson/configuration/serverAtag.jks*	Intel® CIT tag keystore path.
MTWILSON_TAG_KEYSTORE_PASS	mypassword	Intel® CIT tag keystore password.
MTWILSON_TAG_SERVER_PRIVATE	192.168.1.1	The IP Address for the adapter on the private PXE network used for pull provisioning. Not required if pull provisioning is not used.

Table 1. Installation Options (set in *mtwilson.env*) (Continued)

Environment Variable	Default Value or Example	Notes
NO_POSTGRES_MONIT	FALSE	Do not install PostgreSQL configuration for **monit**.
NO_TOMCAT_MONIT	FALSE	Do not install Tomcat configuration for **monit**.
POSTGRES_DATABASE	${DATABASE_SCHEMA}	The schema name for the Attestation Service database. This should not be changed, as it is set by the *DATABASE_SCHEMA* variable.
POSTGRES_HOSTNAME	${DATABASE_HOSTNAME}	The IP Address or hostname of the database server. This should not be changed, as it is set by the *DATABASE_HOSTNAME* variable.
POSTGRES_PASSWORD	${DATABASE_PASSWORD}	The password for the Intel® CIT database. This should not be changed, as it is set by the *DATABASE_PASSWORD* variable.
POSTGRES_PORTNUM	${DATABASE_PORTNUM}	The port number for the Postgres server.
POSTGRES_REQUIRED_VERSION	9.3	The minimum required Postgres version. The value shown is the default.
POSTGRES_USERNAME	${DATABASE_USERNAME}	The user name for the Attestation Service database. This should not be changed, as it is set by the *DATABASE_USERNAME* variable.
PRIVACYCA_DOWNLOAD_PASSWORD		The password for the PrivacyCA user defined in *PRIVACYCA_DOWNLOAD_USERNAME*.
PRIVACYCA_DOWNLOAD_USERNAME		The username used for authenticating Trust Agent hosts. This username and its associated password are entered when installing the Trust Agent on a host so that the Trust Agent can download required certificate information from the Attestation Service server.
TAG_ISSUER_DN	“CN=mtwilson-tag-ca”	Tag issuer distinguished name.
TAG_PROVISION_EXTERNAL	FALSE	If set to **FALSE** (default), the built-in certificate authority is used to sign certificate requests. If set to **TRUE**, certificate requests are not signed automatically, so that the external certificate authority can sign them instead.
TAG_PROVISION_NOCACHE	TRUE	If this is set to **TRUE**, tag provisioning request ignores any existing valid certificates for the specified host and creates a new certificate with the specified selection. If set to **FALSE**, the most recent existing valid certificate is provisioned, and a new certificate is only created if there is no existing valid certificate for the specified host.

Table 1. Installation Options (set in *mtwilson.env*) (Continued)

Environment Variable	Default Value or Example	Notes
TAG_PROVISION_XML_ENCRYPTION_REQUIRED	FALSE	Determines whether XML encryption is required. If this is set to **FALSE**, asset tag provisioning requests using an XML of tag selections can be used without being encrypted. If this value is set to **TRUE**, only encrypted XMLs are accepted.
TAG_PROVISION_XML_PASSWORD	mypassword	XML encryption password.
TAG_VALIDITY_SECONDS	31536000	Tag certificate validity duration in seconds.
TOMCAT_CREATE_SSL_CERT	yes	A **yes** value tells the installer to create a new SSL certificate. A **no** skips the SSL certificate creation. An SSL certificate then needs to be provided.
TOMCAT_REQUIRED_VERSION	7	The minimum required Tomcat version.
TOMCAT_SSL_CERT_CN	${MTWILSON_SERVER}	The Common Name (CN) of the SSL certificate MUST contain the IP Address or the hostname of the Intel® CITserver. Use comma-separated values for multiple CN entries (hostname, IP, etc.). This needs to include the IP Addresses for ALL adapters that communicate with Intel® CIT. Using the value of ${MTWILSON_SERVER}, 127.0.0.1, and the IP Address for the PXE network for Pull asset tag provisioning (if Pull provisioning is used) is recommended.
WEBSERVER_VENDOR	tomcat	The application server to be used. Currently the only accepted value is **tomcat**.

Several users are created during the Attestation Service installation. Each of these users is created with different roles and permissions for different purposes:

- *MC_FIRST_USERNAME*

  This user is created with full Administrator rights. This user can be used to approve new users, as well as any other task with the Attestation Service. Use this user sparingly.

- *PRIVACYCA_DOWNLOAD_USERNAME*

  This user is created with permissions to create and retrieve TPM passwords, to download the PrivacyCA certificate from the Attestation Service, and to create VM attestations. This user is intended for use when installing the Trust Agent.

- *MTWILSON_TAG_ADMIN_USERNAME*

  This user is created with permissions to create host attestations, as well as all permissions for Asset Tagging. This user is recommended for use with the OpenStack Controller for OpenStack deployments, as it has all required permissions for the OpenStack scheduler plugins and other tasks required for integration.

5.9.1.2 Configuration Options

These configuration settings are saved in *.properties* files in */etc/intel/cloudsecurity/*. These files are encrypted by default after installation. These properties can be viewed using the following command:

```
mtwilson export-config --in=<path_to_properties_file> --stdout
```

If no specific *.properties* file is specified, *mtwilson.properties* is used.

To change the value of any configuration setting in *mtwilson.properties*, use the following command:

```
mtwilson config <key> <value>
```

To change the value of any property not in *mtwilson.properties*, decrypt the needed properties file, change the property, and then re-encrypt the file:

```
mtwilson export-config --in=<path_to_encrypted_file> --out=<temporary_filename>
```

Make changes in the temporary file:

```
mtwilson import-config --in=<temporary_filename> --out=<path_to_encrypted_file>
```

Note: The *mtwilson.properties* file is a **master** configuration file. When Attestation Service components start up, properties configured in this file take precedence.

Table 2. *mtwilson.properties*

Configuration Key	Default Value or Example	Notes
dbcp.validation.on.borrow=	true	
dbcp.validation.on.return=	false	
dbcp.validation.query=	select 1	
mtwilson.api.tls.policy.certificate.sha256=	sha256 of the Attestation Service TLS certificate	sha256 of the Attestation Service TLS certificate.
mtwilson.api.url=	https://<*AttestationService IP or Hostname*>\:8443/mtwilson/v1	Base URL for v1 APIs.
mtwilson.as.autoUpdateHost=	false	This setting is for a future feature of the Attestation Service. This setting must be set to **false**.
mtwilson.as.dek=	Randomly-generated password	Decryption key used by the Attestation Service to encrypt sensitive connectivity details in the database, such as vCenter administrator passwords.
mtwilson.privacyca.aik.p12.password=	Randomly-generated password	Password for the Privacy CA Attestation Identity Key (AIK).
mtwilson.privacyca.ek.p12.password=	Randomly-generated password	Password for the Privacy CA Endorsement Key (EK).
mtwilson.tls.keystore.password=	Randomly-generated password	Password to the Attestation Service TLS keystore.
Web Service Settings		
mtwilson.webserver.vendor=	tomcat	Defines the web service application being used.
tomcat.admin.password=	Randomly-generated password	Randomly-generated password for the web service account.
tomcat.admin.username=	admin	Defines the username for a web service account that is used to run Attestation Service components instead of root.

Table 2. *mtwilson.properties* (Continued)

Configuration Key	Default Value or Example	Notes
Database Configuration		
mtwilson.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.
mtwilson.db.host=	127.0.0.1	IP Address or hostname of the database server.
mtwilson.db.password=	password	Database password.
mtwilson.db.port=	5432	Database server port.
mtwilson.db.schema=	mw_as	Database schema name (mw_as by default).
mtwilson.db.user=	root	Database username.
Asset Tag Configuration		
mtwilson.atag.html5.dir=	*/usr/share/apache-tomcat-7.0.34/ webapps/mtwilson-portal/tag*	The path to the asset tag HTML5 information. By default this is: */usr/share/apache-tomcat-7.0.34/ webapps/mtwilson-portal/tag*
mtwilson.atag.mtwilson.baseurl=	https://<*AttestationService Ip or Hostname*>\:8443/mtwilson/v1/ AttestationService/resources/ assetTagCert	The base URL for Asset Tag APIs. This appears in the following format: https://<*MtWilson IP or Hostname*>\:<*MtWilson Port*>/mtwilson/ v1/AttestationService/resources/ assetTagCert
mtwilson.atag.url=	https://<*AttestationService IP or Hostname*>\:8443/mtwilson/v2	The base URL for Asset Tag API resources. The default value is: https://<*AttestationService IP or Hostname*>\:8443/mtwilson/v2.
mtwilson.locales=	en-US	Defines the locale for the Attestation Service language settings. Currently only en-US is supported.
mtwilson.tag.api.password=	Randomly-generated password	Credentials for a background user used by the Asset Tag service.
mtwilson.tag.api.username=	tagservice	Credentials for a background user used by the Asset Tag service.
tag.issuer.dn=	CN\=mtwilson-tag-ca	Distinguished Name to be used in Asset Tag certificates created by the Attestation Service.
tag.provision.autoimport=	true	If set to **true** (default), all asset tag certificates are automatically imported to be matched to an appropriate host for tag attestation. If set to **false**, the certificates are not imported automatically. Attestation of asset tag information depends on the certificate being imported, so Intel recommends that this be set to **true** for must deployments.
tag.provision.external=	false	Use external CA instead of the built-in CA. If set to **true**, certificates are not automatically signed by the Attestation Service CA, and all certificates need to be signed by the external authority instead. Set to **false** by default.
tag.provision.xml.encryption.password=	password	Password to decrypt encrypted XML files.

Table 2. *mtwilson.properties* (Continued)

Configuration Key	Default Value or Example	Notes
tag.provision.xml.encryption.required=	false	Determines whether the Attestation Service accepts Asset Tags selections unencrypted. Set to **false** by default, which enables plain-text XML to be used when generating selections or certificates. If set to **true**, only properly encrypted XML input is allowed.
tag.validity.seconds=	31536000	This defines the validity duration for Asset Tag certificates in seconds.
TLS Policy Configuration		
mtwilson.default.tls.policy.id=	TRUST_FIRST_CERTIFICATE	Sets the default TLS policy. Can be the name or Universally Unique Identifier (UUID) of a specific shared policy. This policy is used for all APIs that do not specify another TLS policy.
mtwilson.tls.policy.allow=	certificate,certificate-digest, public-key,public-key-digest, TRUST_FIRST_CERTIFICATE, INSECURE	Comma-separated list of allowed TLS policy types. Possible values are: • certificate • certificate-digest • public-key • public-key-digest • TRUST_FIRST_CERTIFICATE • INSECURE

Table 3. *audit-handler.properties*

Configuration Key	Default Value or Example	Notes
mountwilson.audit.async	true	Defines whether to do the auditing in a synchronous manner, or not.
mountwilson.audit.db.driver	*org.postgresql.Driver*	JDBC connection driver.
mountwilson.audit.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.audit.db.password=	password	Database password.
mountwilson.audit.db.port=	5432	Database port.
mountwilson.audit.db.schema=	mw_as	Database schema name.
mountwilson.audit.db.user=	root	Database username.
mountwilson.audit.enabled=	true	Enables or disables auditing.
mountwilson.audit.logunchangedcolumns	false	Defines whether to log all the columns or just the changed columns for any updates.

Table 4. *attestation-service.properties*

Configuration Key	Default Value or Example	Notes
Database Configuration		
mountwilson.as.db.driver=	*org.postgresql.Driver*	These settings are used by the attestation service to verify Attestation Identification Key (AIK) quotes. **Do not change these settings.**
mountwilson.as.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.as.db.password=	password	The database connection driver to be used.
mountwilson.as.db.port=	5432	Database schema name (mw_as by default).
mountwilson.as.db.schema=	mw_as	Database username.
mountwilson.as.db.user=	root	Database password.
Trust Agent Quote Verification		
com.intel.mountwilson.as.aikqverify.cmd=	aikqverify	
com.intel.mountwilson.as.home=	*/var/opt/intel/aikverifyhome*	
com.intel.mountwilson.as.openssl.cmd=	openssl.sh	
SAML Configuration		
saml.key.alias=	samlkey1	Alias of the SAML certificate in the SAML keystore.
saml.key.password=	Randomly-generated password	Password for the SAML certificate in the SAML keystore.
saml.keystore.file=	*/etc/intel/cloudsecurity/SAML.jks*	The path to the SAML keystore in *.jks* format.
saml.keystore.password=	Randomly-generated password	Password to the SAML keystore.
saml.validity.seconds=	3600	The length of time, in seconds, that a SAML assertion is considered valid.
Performance		
mtwilson.bulktrust.threads.max=	32	Bulk trust APIs can get attestation reports from multiple hosts simultaneously. This setting controls the maximum number of concurrent threads that the Attestation Service uses. This value is set at 32 threads by default.
Miscellaneous		
com.intel.mountwilson.as.trustagent.timeout	600	Number of seconds that Attestation Service should wait for a response from a Trust Agent.
mtwilson.api.url=	https://<*AttestationService IP or Hostname*>\:8443/mtwilson/v1	The base URL for the v1 APIs.
mtwilson.as.dek	Randomly-generated password	The base64 encoded form of the data encryption key. This key is used to encrypt credentials stored in the database that are used to access vCenter and Citrix Xen APIs.
privacyca.server	Hostname or IP Address	Set the local host's external IP Address or hostname.

Table 5. *wlm-service.properties*

Configuration Key	Default Value or Example	Notes
Database Connection		
mountwilson.as.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.
mountwilson.as.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.as.db.password=	password	Database password.
mountwilson.as.db.port=	5432	Database server port.
mountwilson.as.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.as.db.user=	root	Database username.

Table 6. *management-service.properties*

Configuration Key	Default Value or Example	Notes
mtwilson.api.baseurl	https://attestation.servername:8443/ mtwilson/v1	
Database		
mountwilson.ms.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.
mountwilson.ms.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.ms.db.password=	password	Database password.
mountwilson.ms.db.port=	5432	Database server port.
mountwilson.ms.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.ms.db.user=	root	Database username.
mtwilson.api.baseurl=	https://10.1.71.130:8443/mtwilson/v1	Base URL for the v1 APIs.
mtwilson.api.ssl.policy	TRUST_FIRST_CERTIFICATE	
mtwilson.ms.biosPCRs=	0;17	Configures the default PCRs to be used for a future Attestation Service feature. **DO NOT CHANGE**
mtwilson.ms.bulkmgmt.threads.max	32	
mtwilson.ms.registration.hostTimeout	600	Timeout value for bulk host registration in seconds.
mtwilson.ms.vmmPCRs=	18;19;20	Configures the default PCRs to be used for a future Attestation Service feature. **DO NOT CHANGE.**
mtwilson.privacyca.certificate.file=	*PrivacyCA.p12.pem*	Filename of the Attestation Service Privacy CA certificate in *.pem* format.
mtwilson.rootca.certificate.file=	*MtWilsonRootCA.crt.pem*	Filename of the Attestation Service Root CA certificate in *.pem* format.
mtwilson.saml.certificate.file=	*saml.crt.pem*	Filename of the Attestation Service SAML certificate in *.pem* format.
mtwilson.ssl.required	true	
mtwilson.tls.certificate.file=	*ssl.crt.pem*	Filename of the Attestation Service TLS certificate in *.pem* format.

Table 7. *mtwilson-portal.properties*

Configuration Key	Default Value or Example	Notes
mtwilson.api.baseurl=	https://<*AttestationService IP or Hostname*>:8443/mtwilson/v1	The base URL for the v1 APIs.
mtwilson.api.ssl.policy=	TRUST_FIRST_CERTIFICATE	Name or UUID of the default TLS Policy to be used when a TLS policy is not specified.
Management Console properties		
mtwilson.mc.keystore.dir=	*/var/opt/intel/management-console/users*	Path to users folder for management console user keystores. This property is no longer used as user keystores are now located in the database.
Portal Icon Configuration		
imagesRootPath=	*images/*	Root path for portal icons.
kvm=	*images/kvm.png*	Path to the icon to display for KVM hosts.
suse=	*images/suse.png*	Path to the icon to display for SuSe hosts.
trustFalse=	*images/UnTrusted.png*	Path to the icon to display for untrusted attestations
trustTrue=	*images/Trusted.png*	Path to the icon to display for trusted attestations.
trustUnknown=	*images/Unknown.png*	Path to the icon to display for unknown attestations.
ubuntu=	*images/ubuntu.png*	Path for the icon to display for Ubuntu hosts.
vmware=	*images/vmware.png*	Path to the icon to display for VMware hosts.
xen=	*images/xen.png*	Path to the icon to display for Xen hosts.
Session Timeout Settings		
mtwilson.tdbp.sessionTimeOut=	1800	Duration in seconds for the portal login session to time out and require a new login.
Page Formatting		
mtwilson.tdbp.paginationRowCount=	10	Number of results to display per page.
Trust Dashboard Users Configuration		
mtwilson.tdbp.keystore.dir=	*/var/opt/intel/trust-dashboard/users*	Path to users folder for Trust Dashboard user keystores. This property is no longer used as user keystores are now located in the database.
Whitelist Portal Properties		
mtwilson.wlmp.keyStoreFileName=	*/etc/intel/cloudsecurity/mw.jks*	Deprecated and no longer used.
mtwilson.wlmp.moduleAttestation=	VMWARE	
mtwilson.wlmp.moduleAttestationVersion=	5.1	
mtwilson.wlmp.openSourceHyper-Visors=	KVM;Xen	
mtwilson.wlmp.pagingSize=	8	Number of results to display per page.
mtwilson.wlmp.sessionTimeOut=	1800	Duration in seconds for the portal login session to time out and require a new login.
mtwilson.wlmp.vmwareHypersior=	ESXi	
Database Configuration		
mountwilson.as.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.

Table 7. *mtwilson-portal.properties* (Continued)

Configuration Key	Default Value or Example	Notes
mountwilson.as.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.as.db.password=	password	Database password.
mountwilson.as.db.port=	5432	Database server port.
mountwilson.as.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.as.db.user=	root	Database username.
mountwilson.mc.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.
mountwilson.mc.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.mc.db.password=	password	Database password.
mountwilson.mc.db.port=	5432	Database server port.
mountwilson.mc.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.mc.db.user=	root	Database username.
mountwilson.mcp.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.mcp.db.port=	5432	Database server port.
mountwilson.mcp.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.mcp.db.user=	root	Database username.
mountwilson.ms.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.
mountwilson.ms.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.ms.db.password=	password	Database password.
mountwilson.ms.db.port=	5432	Database server port.
mountwilson.ms.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.ms.db.user=	root	Database username.
mountwilson.tdbp.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.
mountwilson.tdbp.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.tdbp.db.password=	password	Database password.
mountwilson.tdbp.db.port=	5432	Database server port.
mountwilson.tdbp.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.tdbp.db.user=	root	Database username.
mountwilson.wlmp.db.driver=	*org.postgresql.Driver*	The database connection driver to be used.
mountwilson.wlmp.db.host=	127.0.0.1	IP Address or hostname of the database server.
mountwilson.wlmp.db.password=	password	Database password.
mountwilson.wlmp.db.port=	5432	Database server port.
mountwilson.wlmp.db.schema=	mw_as	Database schema name (mw_as by default).
mountwilson.wlmp.db.user=	root	Database username.

5.9.2 Trust Agent - Linux

Table 8. Directory Structure

Item	Default Value	Notes
Application home directory	*/opt/trustagent*	
Configuration directories	*/opt/trustagent/configuration*	
Command-line tools	*/opt/trustagent/bin*	
Java components	*/opt/trustagent/java*	

Table 9. *trustagent.env*

Environment Variable	Default Value or Example	Notes
AIK_INDEX	Default value is 1. Example: 1	
AIK_SECRET	Generated automatically. Example: 024c03d1a2cb95b17a491caf480526f701fc61c7	20 hex-encoded bytes.
JAVA_REQUIRED_VERSION	Default: 1.7 Possible values: Java version number in the format "major.minor.patch_update" Example: 1.7.0_51	Trust Agent for Intel® CIT uses features introduced in Java 7, so the minimum value for this setting is 1.7.
MTWILSON_API_PASSWORD	Example: 94i3kPN7bObnJ5Qo	Password corresponding to *mtwilson.api.username*.
MTWILSON_API_URL	Example: https://10.1.71.56:8443/mtwilson/v2	The URL is for v2 APIs directly, because the 2.x Trust Agent does not use v1 APIs.
MTWILSON_API_USERNAME	Example: admin	Obtain this from the Intel® CIT Attestation Service server; this must be a valid Intel® CIT username with TPM enrollment permission.
MTWILSON_TLS_CERT_SHA256	No default value.	Obtain SHA256 from the Intel® CIT Attestation Service server: *sha256sum/etc/intel/cloudsecurity/ssl.crt*
REGISTER_TPM_PASSWORD=y	REGISTER_TPM_PASSWORD=y	This environment variable is used for registering the TPM Owner Password with the Intel® CIT Attestation Server so that once the asset tag is provisioned, the password is not needed.
TPM_OWNER_SECRET		20 hex-encoded bytes. This can be used to preserve the TPM Owner Secret between installations so that TPM ownership does not need to be reset for a re-installation. ***Note:*** If the OS is re-installed, the file */var/lib/tpm/system.data* contains sealed persistent TPM secrets unique to the TPM owner and needs to be preserved from just before the old OS is wiped and replaced immediately after **trousers** is re-installed in the new OS.

Table 9. *trustagent.env* (Continued)

Environment Variable	Default Value or Example	Notes
TPM_QUOTE_IPV4	Default: true	When enabled (not set or set to **true**) causes the challenger's nonce to be extended with the IP Address of the trusted host for quoting to prevent quote relay attacks. This is a new feature in Intel® Cloud Integrity Technology 2.x; the Intel® Cloud Integrity Technology 1.x equivalent is for this setting to be disabled (set to **false**).
TPM_SRK_SECRET	Constant value. Example: 00000000000000000000	Many tools assume this well-known SRK comprised of 20 zero bytes.
TRUSTAGENT_KEYSTORE_PASSWORD	Generated automatically. Example: 9JF7+HhpMUM_	The password used to access the *trustagent.jk*s file with the keytool.
TRUSTAGENT_LOGIN_PASSWORD	User-specified	By default, the Trust Agent credentials are randomly generated automatically during installation. This optional property allows the administrator to define their own username/ password.
TRUSTAGENT_LOGIN_REGISTER	true	The Trust Agent requires authentication for API access. Setting this option to **true** causes the Trust Agent, during installation, to pre-register its authentication credentials with the Attestation Service. By pre-registering the credentials, no credentials need to be provided by the user when whitelisting or registering the host.
TRUSTAGENT_LOGIN_USERNAME	User-specified	By default, the Trust Agent credentials are randomly generated automatically during installation. This optional property allows the administrator to define their own username/ password.
TRUSTAGENT_PASSWORD	No default value. Possible values: Any combination of printable characters.	The password used to encrypt and decrypt the *trustagent.properties* file. This password must be exported in an environment variable for the Trust Agent to use it.
TRUSTAGENT_TLS_CERT_DN	Possible values: Valid X509 distinguished name with commas separating the parts. For example. CN=trustagent,OU=DCG,O=Intel, L=Folsom,C=US Default: CN=trustagent	Determines the subject name of the Trust Agent's TLS certificate.
TRUSTAGENT_TLS_CERT_DNS	Generated automatically. Comma-separated list of all IP Addresses used by the host.	These names are added as subject alternative names on the Trust Agent's TLS certificate. By default, all names in */etc/hosts* corresponding to local IP Addresses are used. If not specified, the installer performs a reverse DNS lookup for all IP Addresses found in the *ifconfig* output. In some environments this can cause a delay during installation. Manually specifying the subject names can eliminate this delay.
TRUSTAGENT_TLS_CERT_IP	Generated automatically. Possible values: Comma-separated list of IP Addresses.	These addresses are added as subject alternative names on the Trust Agent's TLS certificate. By default, all IP Addresses shown in the *ifconfig* output are used.

Table 10. */opt/trustagent/configuration/trustagent.properties*

Configuration Key	Default Value or Example	Notes
aik.secret=	Generated automatically. Example: 024c03d1a2cb95b17a491caf480526f7 01fc61c7	20 hex-encoded bytes.
binding.key.secret=	dcb272b851a853a0f51d085fb0640a156 8af2fe7	
hardware.uuid=	00886B98-994D-E411-906E-0017A4403562	Host hardware UUID. **Do not change this value.**
mtwilson.api.password=	password	Password corresponding to *mtwilson.api.username*.
mtwilson.api.url=	https://<*Attestation Service IP or Hostname*>\:8443/mtwilson/v2	The URL is for v2 APIs directly because the 2.x Trust Agent does not use v1 APIs.
mtwilson.api.username=	Example: admin	Obtain this from the Intel® CIT Attestation Service server. This must be a valid username with TPM enrollment permission.
mtwilson.tls.cert.sha256=	c5b85d90e13d4258f013023854c9664c0 dc327af	20 hex-encoded bytes. Obtain this from the Attestation Service server: *sha256sum/etc/intel/cloudsecurity/ssl.crt*
signing.key.secret=	68a426fd332058a21189944220da540b 1fbc5710	
tpm.owner.secret=	424c6f724ac9d23f52cb97ef63ec12e62c 893c30	20 hex-encoded bytes. This can be used to preserve the TPM Owner Secret between installations so that TPM ownership does not need to be reset for a re-installation. ***Note:*** If the OS is re-installed, the file */var/lib/tpm/system.data* contains sealed persistent TPM secrets unique to the TPM owner and needs to be preserved from just before the old OS is wiped and replaced immediately after **trousers** is re-installed in the new OS.
tpm.srk.secret=	00000000000000000000	Many tools assume this well-known SRK comprised of 20 zero bytes.
trustagent.keystore.password=	Generated automatically. Example: 9JF7+HhpMUM_	The password used to access the *trustagent.jks* file with the keytool.
trustagent.tls.cert.dn=	CN\=trustagent	This setting determines the subject name of the Trust Agent's TLS certificate.
trustagent.tls.cert.dns=	Generated automatically Comma-separated list of all hostnames used by the host.	These names are added as subject alternative names on the Trust Agent's TLS certificate. By default, all names in */etc/hosts* corresponding to local IP Addresses are used. If not specified, the installer performs a reverse DNS lookup for all IP Addresses found in the *ifconfig* output. In some environments this can cause a delay during installation. Manually specifying the subject names can eliminate this delay.
trustagent.tls.cert.ip=	Generated automatically. Possible values: Comma-separated list of IP Addresses.	These addresses are added as subject alternative names on the Trust Agent's TLS certificate. By default, all IP Addresses shown in the *ifconfig* output are used.

5.9.3 Trust Agent - Windows

Note: The default installation path for Windows Trust Agent is *C:\Program Files (x86)\Intel\TrustAgent*. This path can be changed during installation.

Table 11. Directory Structure

Item	Default Value	Notes
Application home directory	*C:\Program Files (x86)\Intel\TrustAgent*	
Configuration directories	*C:\Program Files (x86)\Intel\TrustAgent\configuration*	
Command-line tools	*C:\Program Files (x86)\Intel\TrustAgent\bin*	
Java components	*C:\Program Files (x86)\Intel\TrustAgent\java*	

Table 12. *trustagent.env*

Environment Variable	Default Value or Example	Notes
AIK_INDEX	Default value is 1. Example: 1	This setting corresponds to the Attestation identity Key index. Do not change this value.
AIK_SECRET	Generated automatically. Example: 024c03d1a2cb95b17a491caf480526f7 01fc61c7	20 hex-encoded bytes.
JAVA_REQUIRED_VERSION	Default: 1.7 Possible values: Java version number in the format "major.minor.patch_update" Example: 1.7.0_51	Trust Agent for Intel® CIT uses features introduced in Java 7, so the minimum value for this setting is 1.7.
MTWILSON_API_PASSWORD	Example: 94i3kPN7bObnJ5Qo	Password corresponding to *mtwilson.api.username*.
MTWILSON_API_URL	Example: https://attestation.server.com:8443/ mtwilson/v2	The URL is for v2 APIs directly, because the 2.x Trust Agent does not use v1 APIs.
MTWILSON_API_USERNAME	Example: admin	Obtain this from the Intel® CIT Attestation Service server; this must be a valid Intel® CIT username with TPM enrollment permission.
MTWILSON_TLS_CERT_SHA256	No default value.	Obtain SHA356 from the Intel® CIT Attestation Service server: *sha256sum/etc/intel/cloudsecurity/ssl.crt*
TPM_OWNER_SECRET		20 hex-encoded bytes. This can be used to preserve the TPM Owner Secret between installations so that TPM ownership does not need to be reset for a re-installation.
TPM_QUOTE_IPV4	Default: true	When enabled (not set or set to **true**) causes the challenger's nonce to be extended with the IP Address of the trusted host for quoting to prevent quote relay attacks. This is a new feature in Intel® Cloud Integrity Technology 2.x; the Intel® Cloud Integrity Technology 1.x equivalent is for this setting to be disabled (set to **false**).
TPM_SRK_SECRET	Constant value. Example: 00000000000000000000	Many tools assume this well-known SRK comprised of 20 zero bytes.

Table 12. *trustagent.env* (Continued)

Environment Variable	Default Value or Example	Notes
TRUSTAGENT_ADMIN_PASSWORD	User-specified	By default, the Trust Agent credentials are randomly generated automatically during installation. This optional property allows the administrator to define their own username/ password.
TRUSTAGENT_ADMIN_REGISTER	true	The Trust Agent requires authentication for API access. Setting this option to **true** causes the Trust Agent, during installation, to pre-register its authentication credentials with the Attestation Service. By pre-registering the credentials, no credentials need to be provided by the user when whitelisting or registering the host.
TRUSTAGENT_ADMIN_USERNAME	User-specified	By default, the Trust Agent credentials are randomly generated automatically during installation. This optional property allows the administrator to define their own username/ password.
TRUSTAGENT_KEYSTORE_PASSWORD	Generated automatically. Example: 9JF7+HhpMUM_	The password used to access the *trustagent.jks* file with the keytool.
TRUSTAGENT_PASSWORD	No default value. Possible values: Any combination of printable characters.	The password used to encrypt and decrypt the *trustagent.properties* file. This password must be exported in an environment variable for the Trust Agent to use it.
TRUSTAGENT_TLS_CERT_DN	Possible values: Valid X509 distinguished name with commas separating the parts. For example. CN=trustagent,OU=DCG,O=Intel, L=Folsom,C=US Default: CN=trustagent	Determines the subject name of the Trust Agent's TLS certificate.
TRUSTAGENT_TLS_CERT_DNS	Generated automatically. Comma-separated list of all IP Addresses used by the host.	These names are added as subject alternative names on the Trust Agent's TLS certificate. By default, all names in */etc/hosts* corresponding to local IP Addresses are used. If not specified, the installer performs a reverse DNS lookup for all IP Addresses found in the *ifconfig* output. In some environments this can cause a delay during installation. Manually specifying the subject names can eliminate this delay.
TRUSTAGENT_TLS_CERT_IP	Generated automatically. Possible values: Comma-separated list of IP Addresses.	These addresses are added as subject alternative names on the Trust Agent's TLS certificate. By default, all IP Addresses shown in the *ifconfig* output are used.

5.10 Security Configuration

The Attestation Service uses the Apache* Shiro* access control framework. Background information on Apache Shiro can be found on the project website (https://shiro.apache.org/web.html). This section describes how Apache Shiro is used in the Attestation Service and how to configure authentication and authorization.

Authentication and authorization in the Attestation Service are configured in the *shiro.ini* file, located in */etc/intel/cloudsecurity*. The default *shiro.ini* configuration, shown below, defines a series of realms and authentication filters, and then applies the filters to various URL patterns. A realm is the source of authentication information, for example the hashed password table. An authentication filter examines each incoming request and determines if it knows how to parse it to obtain the client's credentials and pass them on to the corresponding realm to obtain the authentication information corresponding to the client's credentials. The credentials (authorization token) are evaluated against the authentication information to determine if the password is correct, or if the request signature was can be verified with a known X059 certificate.

Every authentication filter has a chance to look at the request. If no credentials are provided that the filter recognizes, the filter ignores that request and the next filter is checked. However, if credentials are provided that are recognized by the filter but the credentials cannot be validated, the filter rejects the request. This means that when configuring trusted IP Addresses for clients to connect without authentication, if the clients provide incorrect HTTP BASIC or HTTP X509 Authorization credentials, their request is rejected even though it is coming from a trusted IP Address.

Note: All requests from a hostname or IP Address listed in *iniHostRealm.allow* are allowed without any authentication or user role application. This option is commented out by default. Use only with caution.

Following is a sample of *shiro.ini* as configured in the Attestation Service:

```
# shiro configuration
# Reference: https://shiro.apache.org/authentication.html [main]
ssl.enabled = true ssl.port = 8443

jdbcDataSource=com.intel.mtwilson.shiro.jdbi.JdbcDataSource

jdbcPasswordRealm=com.intel.mtwilson.shiro.jdbi.JdbcPasswordRealm
passwordMatcher=com.intel.mtwilson.shiro.authc.password.PasswordCredentialsMatcher
jdbcPasswordRealm.credentialsMatcher=$passwordMatcher

jdbcCertificateRealm=com.intel.mtwilson.shiro.jdbi.JdbcCertificateRealm
certificateMatcher=com.intel.mtwilson.shiro.authc.x509.X509CredentialsMatcher
jdbcCertificateRealm.credentialsMatcher=$certificateMatcher

iniHostRealm=com.intel.mtwilson.shiro.authc.host.IniHostRealm
#iniHostRealm.allow=127.0.0.1
hostMatcher=com.intel.mtwilson.shiro.authc.host.HostCredentialsMatcher
iniHostRealm.credentialsMatcher=$hostMatcher

authcStrategy = com.intel.mtwilson.shiro.LoggingAtLeastOneSuccessfulStrategy

securityManager.realms = $iniHostRealm, $jdbcCertificateRealm, $jdbcPasswordRealm
securityManager.authenticator.authenticationStrategy = $authcStrategy

authcPassword=com.intel.mtwilson.shiro.authc.password.HttpBasicAuthenticationFilter
authcPassword.applicationName=Mt Wilson
authcPassword.authcScheme=Basic authcPassword.authzScheme=Basic

authcX509=com.intel.mtwilson.shiro.authc.x509.X509AuthenticationFilter
authcX509.applicationName=Mt Wilson

# this host filter is an including filter - if any host matches this list it
# will be granted access, but if it doesn't match this list it will be ignored
# and have an opportunity to authenticate with x509 or password
hostAllow=com.intel.mtwilson.shiro.authc.host.HostAuthenticationFilter

[urls]
# the first match wins so order is important
# /path/* will match /path/a and /path/b but not /path/c/d
# /path/** will match /path/a and /path/b and also /path/c/d
# /v2/login is a real resource in mtwilson-shiro-ws-v2 but /v2/logout is a
# virtual resource handled by shiro's logout filter
/v2/login = anon
/v2/logout = logout
/v1/ManagementService/resources/ca/** = ssl
/v1/ManagementService/resources/apiclient/register = ssl
/v1/ManagementService/resources/i18n/locales = ssl
/v2/version = ssl
/v2/rpc/register-user-with-certificate = ssl
/v2/ca-certificates/root = ssl
/v2/ca-certificates/saml = ssl
/v2/ca-certificates/privacy = ssl
/v1/** = ssl, hostAllow, authcX509, authcPassword, perms
/v2/** = ssl, hostAllow, authcX509, authcPassword, perms
/static/** = anon
```

5.10.1 Attestation Service

5.10.1.1 Encrypting the Configuration Files

The Attestation Service configuration files are encrypted by default. The following commands can be used to decrypt or re-encrypt the files. In the samples below, Linux file paths are shown. The same commands are used with the Trust Agent for Microsoft Windows.

1. (Optional) Set the configuration file password (*MTWILSON_PASSWORD*).

   ```
   $ export MTWILSON_PASSWORD="password"
   ```

   This step is optional. By default, this password is randomly generated and configured during the Attestation Service installation. The password is automatically imported when executing Attestation Service commands using either the **root** (for Linux) / **Administrator** (for Windows) or **mtwilson** OS user accounts.

2. Ensure that all changes to be made to the *.properties* files in */etc/intel/cloudsecurity* have already been made before proceeding.

3. Encrypt the files (each *.properties* file must be encrypted and decrypted individually).

   ```
   $ mtwilson import-config --in=/etc/intel/cloudsecurity/attestation-
   service.properties  --out==/etc/intel/cloudsecurity/attestation-
   service.properties
   ```

4. To edit the *.properties* files again later, decrypt them using the following command:

   ```
   $ mtwilson export-config --in=/etc/intel/cloudsecurity/attestation-
   service.properties  --out==/etc/intel/cloudsecurity/attestation-
   service.properties
   ```

The files can now be edited normally, and Step 1 through Step 3 can be repeated to re-encrypt the files. The **ImportConfig** and **ExportConfig** commands have the following additional options:

- To see/export the contents of the encrypted configuration file without replacing it, use the -**-stdout** option:

  ```
  $ mtwilson export-config --in=/etc/intel/cloudsecurity/mtwilson.properties
  --stdout
  ```

- To export the contents of the encrypted configuration file to another file, use the **--out** option:

  ```
  $ mtwilson export-config --in=/etc/intel/cloudsecurity/mtwilson.properties
  --out=outputfilename.properties.
  ```

- To import/encrypt a configuration file from another source like a pipe, use the -**-stdin** option:

  ```
  $ cat editable.properties | mtwilson export-config --out=/etc/intel/cloudsecurity/
  mtwilson.properties --stdin.
  ```

- To import/encrypt a configuration file from another file, use the **--in** option:

  ```
  $ mtwilson import-config --in=inputfilename.properties --out=/etc/intel/
  cloudsecurity/mtwilson.properties --in=inputfilename.properties.
  ```

- (Optional) By default, the *MTWILSON_PASSWORD* variable is used, and this variable is stored in */opt/mtwilson/configuration/.mtwilson_password*. The password is randomly generated during installation, and it is not recommended to change the password. However, if a variable other than *MTWILSON_PASSWORD* is used, users can specify the name of that variable with the **--env-password** option like this:

  ```
  --env-password=MY_OTHER_VARIABLE
  ```

and this can be done for both import and export. Note that the value you supply here is the name of the variable, not the password itself. For example:

```
export OTHER_PASSWORD=changeit

mtwilson import-config --in=/etc/intel/cloudsecurity/mtwilson.properties
--out=/etc/intel/cloudsecurity/mtwilson.properties --env-password=OTHER_PASSWORD

mtwilson export-config --in=/etc/intel/cloudsecurity/mtwilson.properties
--out=/etc/intel/cloudsecurity/mtwilson.properties --env-password=OTHER_PASSWORD
```

- If users do not provide the **--env-password** parameter, the default environment variable *MTWILSON_PASSWORD* is used. If users specify a variable but it is empty, or if users do not specify **-- env-passwor**d and *MTWILSON_PASSWORD* is not set, users are prompted for the password.

```
mtwilson setup ImportConfig /etc/intel/cloudsecurity/mtwilson.properties
```

Note: The Attestation Service Java process needs the encryption password variable resident in memory before the JVM loads. This is handled by default for the *MTWILSON_PASSWORD* variable stored in */opt/mtwilson/configuration/.mtwilson_password*. Custom configuration is needed to ensure any other variable is loaded before the JVM starts on system boot. Otherwise, the variable needs to be manually set before starting the Attestation Service services. Using an alternate variable is supported but strongly not recommended.

- Cannot get password from environment variable 'OTHER_PASSWORD' specified by option 'env-password'

```
You must protect the Mt Wilson Encrypted Configuration File with a password.
Password: Password (again):
```

This means that the value of *MTWILSON_PASSWORD* or its substitute MUST be set before the JVM loads, including at system startup. If this environment variable is not set at system boot before the JVM is started, the Attestation Service services do not start unless manually restarted using the **mtwilson restart** command and the user on the Attestation Service console then is prompted for the password as follows:

```
Cannot get password from environment variable 'OTHER_PASSWORD' specified by option
'env-password'
A password is required to unlock the Mt Wilson Encrypted Configuration File.
Password
```

5.10.1.2 Changing the Database Password

The Attestation Service includes a command-line utility to automatically change the database password and update all related Attestation Service properties files. The command is:

```
mtwilson change-db-pass
```

This utility prompts for the new password, and automatically updates the database password in PostgreSQL as well as all of the *.properties* files. If the *.properties* files have been encrypted, the command automatically handles decryption and re-encryption as long as the appropriate encryption password variable (*$MTWILSON_PASSWORD* by default) has been set. The encryption password is only required if encryption is being used.

5.10.2 Trust Agent

5.10.2.1 Encrypting the Trust Agent Configuration Files

By default, the Trust Agent configuration *.properties* files are encrypted during installation. To view the contents of any Trust Agent *.properties* file, use the following commands:

View only:

```
tagent export-config --stdout
```

Decrypt:

```
tagent export-config --in=<path_to_encrypted_file> --out=<path_to_temp_file>
```

Encrypt:

```
tagent import-config --in=<path_to_temp_file> --out=<path_to_encrypted_file>
```

5.11 High Availability Guidelines

5.11.1 Attestation Service

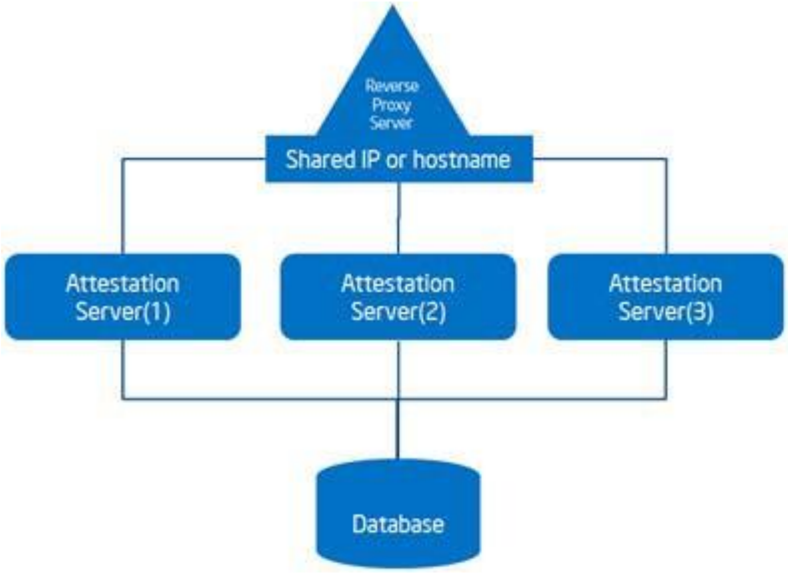
To maximize service availability in an environment where hypervisor-level high availability or fault tolerance solutions are impractical, unavailable, or insufficient, Intel recommends using a redundancy model featuring two (or more) Intel® Cloud Integrity Technology Attestation Server instances running on separate hosts accessing the same external database with traffic directed by a reverse-proxy server.

Note: The shared database server should be configured for redundancy as well. However, database server configuration is beyond the scope of this document.

In this configuration, if one of the Attestation Servers becomes unavailable for any reason, the reverse-proxy automatically fails over to another remaining Attestation Server. Since both instances are accessing a shared database, there is no loss of data in the case of a failover, and there is no manual intervention required after the failed server is brought back on-line.

The instructions below define a sample configuration using the NginX* reverse proxy and two Attestation Server instances. These instructions can be adapted to similar alternative solutions. For example, shared storage or block-level file replication could be used to keep Attestation Server components and configurations synchronized across multiple servers. The key requirements for redundant Attestation Servers are:

- A shared database (whether this is external, or a replicated database across the Attestation Service servers).
- Some form of reverse proxy or other method of sharing a single IP Address or hostname across all Attestation Server instances. If the portal will be used, sessions must be “sticky” or the session information otherwise preserved so that users are not logged out every time they're redirected to a new back-end instance.
- Identical keys, certificates, and other secrets on all Attestation Server instances so that they can all use the same credentials to access the database, and so that the TLS certificate used by the web services remains identical. These include passwords and secrets defined in the various *.properties* files in the Attestation Service configuration directory, all of the certificates and keys in that directory, the TLS certificate for the web server, and the configuration files for the web server.



5.11.1.1 Prerequisites

Following are the prerequisites:

- Separate database server (MySQL or PostgreSQL are supported).
- An additional server running Ubuntu Linux to run the NginX reverse-proxy service (other solutions are also possible).

5.11.1.2 Deployment Instructions

Following are the deployment instructions:

1. Deploy two (or more) separate Attestation Server instances on separate host servers.

   Use a shared IP Address or hostname for the Attestation Server IP when setting up all Attestation Servers. The first installation performed configures the secrets, certificates, keys, and users that are later copied to all of the other instances. Subsequent installations throws errors during user creation, as they attempt to create users that already exist; this is normal and expected.

2. In the *mtwilson.env* installation answer file, configure the IP Address, port, and logon credentials for the external database server as well as the port and logon credentials and the database name.

   Remember, the database schema and a valid user with rights over this database must be created on the database server prior to running the Attestation Server installation, and must be configured to enable connections from all redundant Attestation Servers. Refer to Section 5.6 for remote database configuration instructions. The Attestation Server installer automatically creates the required database tables on the remote server.

3. Copy the SSL certificate from the first Attestation Server to the other(s).

   /opt/mtwilson/share/apache-tomcat-7.0.34/ssl/.keystore

4. Copy the full contents of the */etc/intel/cloudsecurity/* folder from the first Attestation Server to the other(s).

5. After deploying each of the Attestation Server instances, verify that the portals for each are functioning.

6. On the reverse-proxy server, run the following command:

```
$ apt-get install nginx
```

7. Create or modify the *default.conf* configuration file:

```
$ nano /etc/nginx/conf.d/default.conf
```

Edit the file as follows:

```
upstream mtwilson {
    server <Mt Wilson Server 1 IP>:<port>max_fails=1 fail_timeout=30s;
    server <Mt Wilson Server 2 IP>:<port>backup;
}

server {
    listen       <IP Address of NginX Server>:<port>;
    server_name  <DNS Name of NginX Server>;

    access_log  /var/log/nginx/mtwilsonHA.log;
    error_log   /var/log/nginx/mtwilsonHA.error.log;

    ## send request back to mtwilson ##
    location / {
        proxy_pass http://mtwilson/;
        proxy_next_upstream error timeout invalid_header http_500 http_502
        http_503 http_504;
    }
}
```

Consider the following:

— The upstream section allows for the declaration of the Attestation Servers to be used with the reverse proxy, as well as configuration of when each of the servers is used. In this configuration, one server is set to be the primary, and a second server is set to be the backup. Note that all traffic goes to the primary server, and the server labeled backup is only used if the other server has been flagged as failed. This is done to preserve session information when using the Attestation Service portal.

— A failure is triggered when an HTTP request times out. In this case, the *max_fails=1* variable sets a single timeout to trigger a failure. This can be set higher, but doing so increases the amount of time before a failure is detected and remediated. The *fail_timeout=30s* variable tells NginX to leave a server flagged as failed for 30 seconds before trying again. Increasing this value can improve performance slightly when a failover has occurred (as every new request more than 30 seconds since the last failed flag re-attempts the primary server and thus has to wait for a timeout), but also delay the automatic fail-back to the primary server once it is recovered.

— As an alternative to the active/passive configuration, the *ip_hash* line can be added to the beginning of the upstream section. Note that if *ip_hash* is used, remove the *max_fails*, *fail_timeout*, and *backup* arguments. This causes NginX to direct incoming traffic to an Intel® CIT server in the upstream list based on a hash of the requesting machine's IP Address. In this way, NginX can perform load balancing. Requests are divided evenly across all servers in the list, or the weight argument can be added to determine load ratios.

— The *listen* and *server_name* variables tell NginX which specific IP and port to listen on and forward. Requests sent to other ports are not forwarded and receives a default NginX page. This can be configured to be a different page if desired. For more information, refer to the documentation for NginX at http://nginx.org/en/docs/.

— The *proxy_pass* setting tells NginX to forward all requests matching the syntax `http://<listening IP/ name and port>/*` to the Attestation Server setting declared in the upstream section.

8. Restart the NginX service.

```
$ service nginx restart
```

At this point, it should be possible to point a browser to the Attestation Server portal URLs, but substituting the Attestation Server IP/DNS name with the NginX server's IP/DNS name. NginX automatically forwards the URLs to the appropriate Attestation Server, and the services can be used normally.

5.11.1.3 Failover

If a failover to the backup server should occur, there is a brief waiting period (approximately 5-10 seconds) as NginX waits for the timeout and flags the primary as failed. If a user is in the middle of accessing the web portals during this time, they are redirected to the login screen and receive an error that says their session has timed out. Once they log in again, Attestation Server begins functioning normally.

6.0 Uninstallation

6.1 Attestation Service

To remove all data from the database, execute the following command:

```
mtwilson erase-data
```

To delete all users from the database, execute the following command:

```
mtwilson erase-users --all
```

Note: Removing the database contents and user information from the Attestation Service is irreversible.

To uninstall Attestation Service executables and configuration, execute the following command:

```
mtwilson uninstall
```

6.2 Trust Agent

Warning: Re-installing the Trust Agent generates a new binding key, if the VM Privacy use case is enabled. Be sure to re-register the Trust Agent host in the Attestation Service after re-installation to ensure the binding key is also updated.

6.2.1 Uninstalling the Linux Trust Agent

To fully uninstall the Trust Agent and all subcomponents, execute the following command:

```
tagent uninstall
```

If the Trust Agent is re-installed after uninstallation, be sure to delete and re-register the host in the Attestation Service to ensure that all keys and signatures are updated.

6.2.2 Uninstalling the Windows Trust Agent

The Windows Server 2012 Trust Agent can be uninstalled from **Control Panel > Program Files**. Windows Hyper-V server can be uninstalled by executing:

```
Uninstall.exe
```

from the directory in which the Windows Trust Agent was installed. Executing the following command also un-installs the Windows Trust Agent:

```
tagent uninstall
```

7.0 Troubleshooting Guide

The following commands can be used to assist in any troubleshooting.

Table 13. Troubleshooting Commands

Command / Path	Description
/op/mtwilson/share/apache-tomcat-7.0.34/logs/catalina.out (Tomcat)	Path for Attestation Server service log file.
/tmp/mtwilson-install.log	Path for Attestation Server installer log file. ***Note:*** Default is configured by INSTALL_LOG_FILE in the *mtwilson.env* file).
`$ mtwilson fingerprint`	Display the keys and certificates.
`$ mtwilson setup`	Reconfigure the IIntel® CIT certificates and credentials. ***Note:*** Only if previously deleted. Does not overwrite existing.
`$ mtwilson status`	Check the status of the Intel® CIT processes.
`$ mtwilson tomcat-restart`	Restart the Tomcat application server.
`$ mtwilson tomcat-stop`	Stop the Tomcat application server.
`$ mtwilson uninstall`	Uninstall Intel® CIT.
`$ mtwilson version`	Check the currently running Intel® CIT version.
`$ mtwilson-portal setup`	Reconfigure only the admin credentials for the portal.
`$ reboot`	Restart the Intel® CIT virtual machine.

8.0 TXT/TPM Prerequisites and Activation

Detailed information on supported server platforms, white papers, and activation guides can be found at:

http://intel.com/txt

The server support matrix is an up-to-date list of server models that fully meet all Trusted Computing requirements. In some cases, these servers need to be specifically ordered with a Trusted Platform Module. In other cases, a TPM may be included by default. Consult your OEM vendor to ensure your servers include a TPM.

TPMs can be retrofitted to existing servers. Consult your server OEM vendor to determine if this is supported by your OEM and for instructions.

TPMs must be correctly provisioned by the server OEM to function properly. Installing a TPM purchased directly from a third party to retrofit is not recommended.

8.1 Trusted Boot Provisioning (TPM 1.2)

Once TXT/TPM are enabled and activated, the host OS must be configured for Trusted Boot. By default, the underlying TXT/TPM hardware measures the BIOS MLE PCRs (0-17). However, PCRs 18 and up require the following:

- Booting using the **tboot** package.
- The **trousers** package to talk to the TPM.
- The Intel® Cloud Integrity Technology patched version of the TPM Tools package in order to work correctly with the Trust Agent. This is a prerequisite step for configuring the Trust Agent on any Linux host.

Notes: The *install-patched-tpm-tools.bin* tool is needed for Linux Nodes running Trust Agent.

The *txtprov-ubuntu.bin* tool is a reference script that configures an Ubuntu host to boot using Tboot.

8.1.1 Linux (TPM 1.2)

1. Install *virt-manager,qemu-kvm*. (Optional. Needed only if the host is used as a KVM hypervisor.)
2. Install **tboot** 1.8.1 or higher (for Ubuntu use Intel® CIT-provided *TXTProv-Ubuntu.bin* installer package).
3. Install *trousers_0.3.13-3* or higher package.
4. Run the Intel® CIT-provided *install-patched-tpm-tools.bin* tool.
5. Reboot. Ensure that as the system boots, the boot menu uses the **tboot** option.
6. After the OS loads, use the **txt-stat** command to verify that the system has successfully launched in Trusted Boot mode. If successful, at the top of the output of **txt-stat** you should see:

```
"TXT measured launch: TRUE"
```

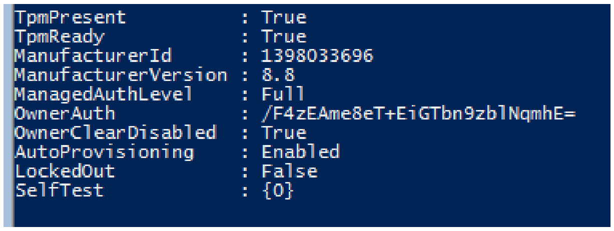
8.1.2 Linux (TPM 2.0)

1. Enable TPM 2.0 Drivers in the OS.
2. Install **tboot** 1.9.4 or higher.
3. Change the grub boot loader to make Tboot the default boot option.
4. (Optional. Needed only if the host will be used as a KVM hypervisor.) Install *virt-manager,qemu-kvm*.
5. Reboot. Ensure that as the system boots, the boot menu uses the **Tboot** option.
6. After the OS loads, use the **txt-stat** command to verify that the system has successfully launched in Trusted Boot mode. If successful, at the top of the output of **txt-stat** you should see:

   ```
   "TXT measured launch: TRUE"
   ```

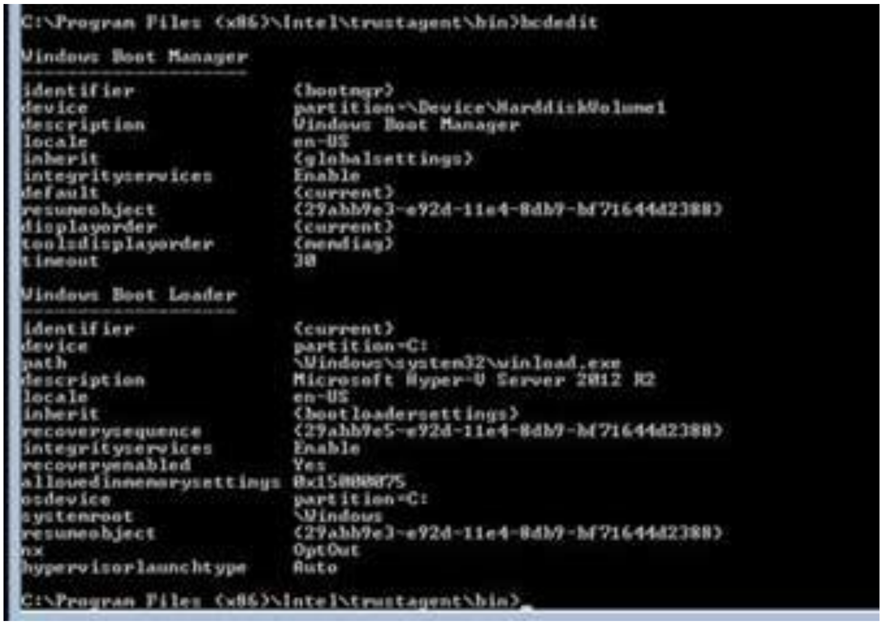
8.1.3 Microsoft Hyper-V 2012 Server (TPM 1.2/2.0)

1. Clear and Enable TPM. Enable Intel® TXT.
2. Execute **dism /online /enable-feature /FeatureName:tpm-psh-cmdlets** in powershell.
3. Execute **get-tpm** and TPMPresent and TPMReady status should be: **True**



```
TpmPresent          : True
TpmReady            : True
ManufacturerId      : 1398033696
ManufacturerVersion : 8.8
ManagedAuthLevel    : Full
OwnerAuth           : /F4zEAme8eT+EiGTbn9zblNqmhE=
OwnerClearDisabled  : True
AutoProvisioning    : Enabled
LockedOut           : False
SelfTest            : {0}
```

4. If not, execute: **Initialize-tpm**
5. Check the output of the **get-tpm** for **True** status.
6. In the command line, execute the following:
   a. `bcdedit /set {bootmgr} integrityservices enable`
   b. `bcdedit /set  integrityservices enable`
7. Ensure that the **bcdedit** settings are set:
   - Windows Boot Manager: Integrityservices is set to enable.
   - Windows Boot Loader: intergrityservices is set to enable.



```
C:\Program Files (x86)\Intel\trustagent\bin>bcdedit

Windows Boot Manager
--------------------
identifier              {bootmgr}
device                  partition=\Device\HarddiskVolume1
description             Windows Boot Manager
locale                  en-US
inherit                 {globalsettings}
integrityservices       Enable
default                 {current}
resumeobject            {29abb9e3-e92d-11e4-8db9-bf71644d2388}
displayorder            {current}
toolsdisplayorder       {memdiag}
timeout                 30

Windows Boot Loader
-------------------
identifier              {current}
device                  partition=C:
path                    \Windows\system32\winload.exe
description             Microsoft Hyper-V Server 2012 R2
locale                  en-US
inherit                 {bootloadersettings}
recoverysequence        {29abb9e5-e92d-11e4-8db9-bf71644d2388}
integrityservices       Enable
recoveryenabled         Yes
allowedinmemorysettings 0x15000075
osdevice                partition=C:
systemroot              \Windows
resumeobject            {29abb9e3-e92d-11e4-8db9-bf71644d2388}
nx                      OptOut
hypervisorlaunchtype    Auto

C:\Program Files (x86)\Intel\trustagent\bin>_
```

8. Run the installer:

```
mtwilson-trustagent-windows-installer-3.0-SNAPSHOT.exe
```

```
C:\Users>mtwilson-trustagent-windows-installer-3.0-SNAPSHOT.exe
```

8.1.4 Microsoft Windows Server 2012 (TPM 1.2/2.0)

No additional steps are required to enable trusted boot for Microsoft Windows Server 2012. The TPM must be owned by the Microsoft Windows operating system; this can be redone by clearing the TPM ownership and reactivating TPM/TXT. If the TPM ownership password is known, the **tpm.msc** utility can be run.

https://technet.microsoft.com/en-us/library/cc749022(v=ws.10).aspx

This utility allows ownership to be taken (if the password is known), cleared, and so on.

9.1 Frequently Asked Questions

What is the process involved to upgrade the BIOS/VMM/OS of the measured host?

When upgrading the BIOS, OS, or hypervisor versions, import new MLEs from a previously upgraded known good host as per the normal Whitelisting guidelines. As each other registered host is upgraded, either edit the host in the Attestation Server (**Host Management > Edit Host**) and re-select the MLEs used as appropriate, or delete the host (**Most Management > Delete Host**) and repeat the registration process, which automatically selects the correct MLEs if available.

Does the Intel® CIT Attestation Server require an Internet connection?

The Intel® CIT Attestation Server installer needs access to package repositories so that prerequisite packages can be installed. An Internet connection is not required by Intel® CIT after installation.

Does Intel® CIT need any firewall exceptions?

Intel® CIT requires the ports used by the web server (8443 for Tomcat) be open. If your configuration includes a remote database, the database port must be open as well (5432 for PostgreSQL, by default). The Trust Agent communicates over port 1443 by default.

Can we have a whitelist server in a production environment?

Intel strongly recommends configuring all whitelist servers in an isolated, non-production environment, but this is not a requirement. After the whitelist MLEs are configured, the server(s) used to generate the MLEs can be moved to production.

Does the Intel® CIT Attestation Server database use a single-tenant model?

Yes, the Intel® CIT database uses a single-tenant model.

What non-Intel® CIT services are used in the Intel® Cloud Integrity Technology 2.2 release?

- Tomcat 7.0.34
- PostgreSQL 9.3
- Monit
- Logrotate

What are the minimum required versions for the database and web server components?

Intel® Cloud Integrity Technology 2.2 requires Tomcat 7.0.34 as well as PostgreSQL 9.3 as minimum versions.

How can I restart the Intel® CIT Attestation Server services if the services are not responding?

If the Intel® CIT Attestation Server is not responding, restart the JVM processes using the following command:

```
$ mtwilson restart
```

Does the Attestation Server feature high availability?

Yes, the Attestation Server supports high availability. If the application server hangs or otherwise fails, it automatically restarts without additional manual input by the **monit** process monitoring service. This functionality requires the monit option in the *.env* file during installation. Additional functionality is provided by VMware High Availability. The Attestation Server can also be configured in an Active/Passive or Active/Active mode using a shared database. For additional options, refer to Section 5.11, “High Availability Guidelines” and the documentation for your specific hypervisor.

What are the password requirements for Intel® CIT?

Intel® CIT does not have any internal requirements for password length or complexity. However, the database servers (MySQL and PostgreSQL) do have special character restrictions. The following characters cannot be used in a database password:

/ \ | '

If using PostgreSQL for the Intel® CIT database, the following additional characters cannot be used:

& * ()

Can the database password be changed?

The database passwords can be changed at any time. A command-line script is included with tIntel® CIT to automate this process, and automatically changes the database password and update the relevant Intel® CIT configuration files.

```
$ mtwilson change-db-pass
```

The script prompts for a new password, and automatically updates the database password in the existing configuration files and the database itself. If the properties files have been encrypted (refer to Section 5.10 for details on encrypting the properties files), the $MTWILSON_PASSWORD environment variable needs to be set with the encryption password before running the **change-db-pass** command.

The database password can also be changed manually. Users need to change the password for the database user directly in MySQL or Postgres, update each of the **db.password** parameters in the *.properties* files in */etc/intel/cloudsecurity*, and restart the Intel® CIT services. The manual commands for changing passwords in MySQL and PostgreSQL are:

For MySQL:

```
$ dpkg-reconfigure mysql-server-5.1 # enter desired password
```

For Postgres:

```
$ psql PostgreSQL
> alter user username with password 'new_password';
> \q
```

Is TPM portable?

Yes, TPM is currently portable.

Current behavior:

- In ESXi:
  - — TPM portability is possible only if the TPM is manually cleared in the BIOS console. Otherwise, the TXT launch is not successful at boot.
  - — PCR 0 remains the same after TPM replacement. This is due to TPM clear and re-provision.
- In Open Source Linux:
  - — TPM portability is possible across two servers.
  - — PCR 0 changes after TPM replacement. PCR 0 is extended with previous hash value so a new set of PCR 0 is derived.
    - TPM Clear > TPM off > TPM On operation is required from the BIOS console to recover the actual PCR 0 value.

NOTE: This page intentionally left blank.

LEGAL

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors which may cause deviations from published specifications.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.