



Intel® Authenticate

Attestation Guide

Version 3.8

Document Release Date: 30 May 2019

Legal Notices and Disclaimers

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel, Intel vPro, Intel Core, Xeon, and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Intel is under license.

© 2016-2019 Intel Corporation

1 Introduction

This document describes how to perform attestation (verification) of Intel® Authenticate.

Intel recommends performing the initial installation and configuration of Intel Authenticate in a controlled and secure environment. However, in most situations, you will install and configure remotely on platforms that are already deployed “in the field” to your users. Intel Authenticate cannot authenticate the administrator who performs the initial configuration and therefore does not conduct any checks during the initial configuration stage. But all subsequent administrative actions (for example, setting a new policy) are permitted only if they are authorized under the credentials (certificate) set in the initial configuration. This still means that, theoretically, the platform could have been compromised before the initial configuration.

The Attestation Utility enables you to perform three types of attestation:

- **Environment** – Verify that Intel Authenticate is actually running in the Trusted Execution Environment (TEE) of the Intel® Management Engine (Intel® ME). These tests are done using Intel® Enhanced Privacy ID (Intel® EPID). Periodically, you will need to update the certificate revocation lists (see [Updating the Intel® EPID Files](#) on page 6.)
- **Data** – Verify that the Intel Authenticate data stored in the Intel ME is the correct data that was configured by your back-end infrastructure. These tests are done by comparing the data with expected values. For example, the policy that you defined in the back-end and that you expect is configured on the platform.
- **Process** – Verify that the response sent from the platform to the back-end really was sent from Intel Authenticate and not by an imposter. This test is done by comparing a nonce returned from the platform is identical to the nonce that was sent to the platform. (A “Nonce” is a random sequence of numbers used in cryptography for verification purposes.)

Note:

- The Attestation Utility supports a single direction mechanism where Intel Authenticate can attest the correctness of the environment and data to the back-end. This mechanism does not attest the identity of the back-end to Intel Authenticate.
- The Attestation Utility is not supported when deployment is performed via McAfee ePO.

Attestation involves three steps:

1. In the back-end, build an attestation request (see [Building the Attestation Request](#) on the next page).
2. On the platform, get the attestation data (see [Getting the Attestation Data](#) on page 3).
3. In the back-end, verify the attestation response (see [Verifying the Attestation Response](#) on page 4).

Note:

It is possible to perform all three steps on the platform. But it is recommended to always perform step 1 and step 3 in a secure environment in the back-end.

2 Building the Attestation Request

To create the attestation request, use the `Build-Request` command of the Attestation Utility.

Note:

The Attestation Utility can only be run on 64-bit operating systems, and requires Microsoft Visual C++ 2015 Redistributables (or higher) to be installed.

This is the syntax (not case sensitive):

```
AttestationUtility.exe Build-Request [/RequestFile <file path>] [/?]
```

Flag	Details
<code>/RequestFile <file path></code>	The file in which to save the attestation request. If not supplied, the request is saved in a file named <code>AttestationReq.bin</code> in the same folder as the executable.
<code>/?</code>	Show help

To build the attestation request:

1. Copy the `Attestation > Backend` folder to a computer in your back-end. For example, the server from which you deployed Intel Authenticate to the client platforms.
2. In the `Backend` folder, open a command prompt as an administrator.
3. Type in the `Build-Request` command. For example:

```
AttestationUtility.exe Build-Request
```

4. Press `<Enter>`. When complete, a "SUCCESS" message is displayed and the `AttestationReq.bin` file is created.

Note:

In addition to the attestation request file, a file named `nonce.txt` is also generated. The `nonce.txt` file contains the nonce that is included as part of the attestation request. You can supply this file when verifying the response data.

3 Getting the Attestation Data

To get the attestation data from the platform, use the `Get-Attestation.ps1` PowerShell script.

To get the attestation response:

1. Move the `AttestationReq.bin` (or the name of the file that you defined) to the `Attestation > Client` folder.
2. Copy the `Client` folder to the platform on which you want to perform attestation.
3. In the `Client` folder, open a PowerShell command prompt as an administrator.
4. Run the `Get-Attestation.ps1` script and supply the name of the attestation request file. For example:

```
.\GetAttestation.ps1 .\AttestationReq.bin
```
5. Press <Enter>. When complete, a "Succeeded" message is displayed and the `AttestationData.bin` file is created.

4 Verifying the Attestation Response

To verify the attestation data, use the `Verify-Response` command of the Attestation Utility.

Note:

The Attestation Utility can only be run on 64-bit operating systems, and requires Microsoft Visual C++ 2015 Redistributables (or higher) to be installed.

This is the syntax (not case sensitive):

```
AttestationUtility.exe Verify-Response [/AttestationFile <file path>]
[/EpidFiles <file path>] [/NonceFile <file path>] {[PolicyFile <file path>]
| [SpolFile <filepath>] [/CertFile <filepath>]}
[/ResultsFile <file path>] [/?]
```

Flag	Details
/AttestationFile <file path>	The attestation data file that was generated by the <code>Get-Attestation.ps1</code> script on the platform. If not supplied, the default filename is used (<code>AttestationData.bin</code>).
/EpidFiles <file path>	The folder where the EPID revocation lists and other certificate files are located. If not supplied, the default location is used (the <code>EpidFiles</code> sub folder of the <code>Backend</code> folder).
/NonceFile <file path>	The nonce file that was generated when running the <code>Build-Request</code> command
/PolicyFile <file path>	The Intel Authenticate XML policy file that you expect is configured on the platform
/SpolFile <filepath>	The Intel Authenticate policy in <code>.spol</code> format. Cannot be used with the <code>/PolicyFile</code> flag.
/CertFile <file path>	The Intel Authenticate admin credentials certificate file. Cannot be used with the <code>/PolicyFile</code> flag.
/ResultsFile <file path>	The file in which to save the verification results. If not supplied, the results are saved in a file named <code>Results.json</code> .
/?	Show help

To verify the attestation response:

1. Copy the `AttestationData.bin` file, that was generated by the `Get-Attestation.ps1` script, from the platform to the `Backend` folder where you generated the build-request.
2. In the `Backend` folder, open a command prompt as an administrator.

3. Type in the Verify-Response command. For example:

```
AttestationUtility.exe Verify-Response /NonceFile nonce.txt /PolicyFile
MyPolicy.xml
```

4. Press <Enter>. When complete, a success or failure message is displayed, and the verification results file is created. (In this example, the default file named `Results.json` is created.)
5. Right-click the verification results file and open it with a text editor.
6. Verify the results. This table describes the content of the verification results file.

Entry	Details
<code>appletIdStatus</code>	Verifies that the ID of the Intel Authenticate applet in the Intel ME is the expected ID. A result of "Invalid" means that the installation of Intel Authenticate might be compromised.
<code>dataStatus</code>	Verifies that the data stored in the Intel ME is the correct data that was configured by your back-end infrastructure. A result of "Invalid" means that at least one of the individual data tests has returned a result of "Invalid".
<code>environmentStatus</code>	Verifies that Intel Authenticate is actually running in the Trusted Execution Environment of the Intel ME. A result of "Invalid" means that the trusted environment might be compromised.
<code>nonceStatus</code>	Only tested if you supplied the <code>/NonceFile</code> flag. A result of "Invalid" means that the response sent from the platform does not match the request sent in the Build-Request.
<code>policyHashStatus</code>	Only tested if you supplied the <code>/PolicyFile</code> flag or the <code>/SpolFile</code> flag. A result of "Invalid" means that the Intel Authenticate policy configured in the platform is not the expected policy.
<code>policyVersionStatus</code>	Only tested if you supplied the <code>/PolicyFile</code> flag or the <code>/SpolFile</code> flag. A result of "Invalid" means that the Intel Authenticate policy configured in the platform is not the expected policy.
<code>processStatus</code>	Only tested if you supplied the <code>/NonceFile</code> flag. A result of "Invalid" means that the response sent from the platform does not match the request sent in the Build-Request.
<code>pubKeyHashStatus</code>	Only tested if you supplied the <code>/PolicyFile</code> flag or the <code>/CertFile</code> flag. A result of "Invalid" means that the Intel Authenticate admin credentials set on the platform are not the expected credentials.
<code>trustIdStatus</code>	Not implemented (will always return a result of "Ignored")
<code>version</code>	The schema version of the attestation data (the current version is 1). This entry is for informational purposes only.

5 Updating the Intel® EPID Files

Intel EPID enables Intel platforms to anonymously prove that they are valid members in a group, without revealing any information about their identity. This table describes the main entities of Intel EPID.

Entity	Description
Authority	<p>In Intel EPID, Intel is the “authority”. The authority is responsible for generating the group keys, member keys, parameters, and maintaining revocation lists. These are the two types of keys that are used by Intel EPID:</p> <ul style="list-style-type: none"> • Intel EPID Private Key – Each Intel platform that supports Intel EPID has a unique private key. The private key is burned into the Intel ME by Intel during the manufacturing process. The key is protected by the hardware and cannot be accessed or altered by malware. Each key has one corresponding Intel EPID Group Public Key. • Intel EPID Group Public Key – These keys are also created by Intel, with each public key corresponding to multiple Intel EPID private keys. The public keys do not contain any secrets and are digitally signed by Intel. <p>Note: An attestation mechanism must support the capability to revoke members or groups when necessary. Intel maintains and publishes revocation lists for Intel EPID.</p>
Platform	An Intel platform with, in this case, Intel Authenticate installed. If the platform has a valid Intel EPID private key, and is not listed in a revocation list, then it can prove it is a valid member of an Intel EPID group.
Verifier	The verifier is the entity that tries to establish that the platform is a valid member of the group to which it claims it belongs. In this case that is the Attestation Utility.

In the Intel EPID ecosystem, the authority (Intel) is the only entity that has the privilege to revoke a member or a group. This is done using Certificate Revocation Lists (CRLs):

- `EPID.crl` – This file includes the Group ID revocation list
- `product_XXXXXXXXXX.crl` – These files include private CRLs for each group

The `EpidFiles` folder contains the CRLs that were published by Intel at the time of release of this version of the Attestation Utility. Intel does not provide an automated download or update mechanism for the Intel EPID revocation lists. Organizations that want to use Intel EPID are responsible to check for and download updated revocation lists themselves. You can get the latest CRLs from here:

<https://trustedservices.intel.com/content/crl/>.

The `EpidFiles` folder also includes `.cer` files containing the Intel EPID group public keys. If the `.cer` files in the `EpidFiles` folder do not contain the platform’s group public key, the Attestation Utility will automatically try to download updated `.cer` files. To do this the Attestation Utility requires access to the Internet so that it can contact the Intel EPID Online Verification Service. If Internet access is not available, an error is displayed asking you to run the `Verify-Response` command again when connected to the Internet.