Cisco Secure Boot

This appendix briefly describes the Cisco Secure Boot process and how it impacts image naming conventions.

It contains the following sections:

Fundamental Concepts

Digital signing involves creating a unique digital signature for a given block of data such as software code (often called code or image signing). The signature is created utilizing a hashing algorithm similar to a checksum. Software code can be signed this way and checked at runtime to validate it has not been changed. Typically the code gets a signature calculated by the code owner and this signature is then stored on the system with the code. When the code later executes, it can self validate by using the same algorithm to create its own signature and compare to the pre-computed stored signature, or some other system element can do this signature calculation and check.

A Trusted Element in the scope of system software is a piece of code that is known to be authentic. Trusted code is either immutable (stored in such a way to prevent modification) or sufficient validation mechanisms are in place to insure its authenticity.

The Root of Trust is the lowest layer of the system at which a guaranteed trusted element exists. If the first code executed on systems is immutable, it becomes the Root of Trust in that system.

A Chain of Trust is a series of Trusted Elements whereby each element in the chain is validated as "trusted" by the element before it. A Chain of Trust starts with a Root of Trust element, which validates successive element in the chain, and so on.

Secure Boot Overview

Cisco Secure Boot places the Root of Trust in a hardware chip device on a circuit card where it cannot be changed. The first code (microloader) that executes immediately after power on is guaranteed to be legitimate code from Cisco and programmed during the time of system manufacturing. Furthermore, all software images can be cryptographically verified against modifications prior to load/execution.

The goal of Cisco Secure Boot technology is to address potential issues associated with unprotected boot code.

Once a piece of code is validated, it can be trusted and is allowed to assume control of the processor. Each step of the boot sequence verifies the next step of the boot module via a code-signed module (Chain of Trust).

MIO2 Support for Secure Boot

The ASR 5500 MIO2 supports Secure Boot with a digitally signed image having a Release key. Production MIO2 cards require an image filename signed with a Release key suffix of **.SPA**. For example, asr5500-21.0.0.bin.SPA

Important MIO, DPC and DPC2 cards will also have digitally signed boot images, but they will ignore the signature.

Image Naming Conventions

To distinguish signed from unsigned images, Release Engineering adds suffixes to build names for images that are signed. For example, asr5500-20.0.0.bin.**SPA** indicates a Release key signed as deployable in a customer network.

Verifying Authenticity

The Exec mode **show software authenticity** command displays information about the chain of trust and authentication process for starfile images.

The syntax for this commend is:

show software authenticity { file *url* **[validate] | keys | running }**

Notes:

- **file** *url* **[validate]** displays authenticity information for starfile images on flash or over the network. The **validate** option performs digital signature validation of the image.
- **keys** displays public StarOS key information for each of the key storage regions (Primary, Backup), as well as Rollover key information.
- **running** displays information about the chain of trust for all running software images: StarOS, CFE (bootstrap), BIOS/UEFI (Unified Extensible Firmware Interface) and the microloader.

For additional information about this command, see the *Command Line Interface Reference*.