Trustworthy Systems

Trustworthy Systems

Cisco trustworthy technologies offer product assurance and foundational security capabilities, thereby enhancing the security and resilience of Cisco solutions. To safeguard against device counterfeiting and malicious attacks on hardware and software, Cisco verifies the authenticity and integrity of its solutions using digitally signed software images, hardware-anchored secure boot, Secure Unique Device Identifier (SUDI), and various other trustworthy technologies. Beyond other capabilities, trustworthy technologies conduct automated integrity checks on hardware and software, capable of shutting down the boot process if a compromise is found. The Cisco Trust Anchor module offers a Secure Unique Device Identifier, highly secure storage, a random bit generator, and robust secure key management.

These additional layers of security offer protection against counterfeiting and software modification, facilitate secure and encrypted communications, and confirm that Cisco network devices are functioning as intended.

In trustworthy systems, trust originates at the lowest hardware levels and extends through the boot process, into the operating system (OS) kernel, and ultimately into the OS's runtime

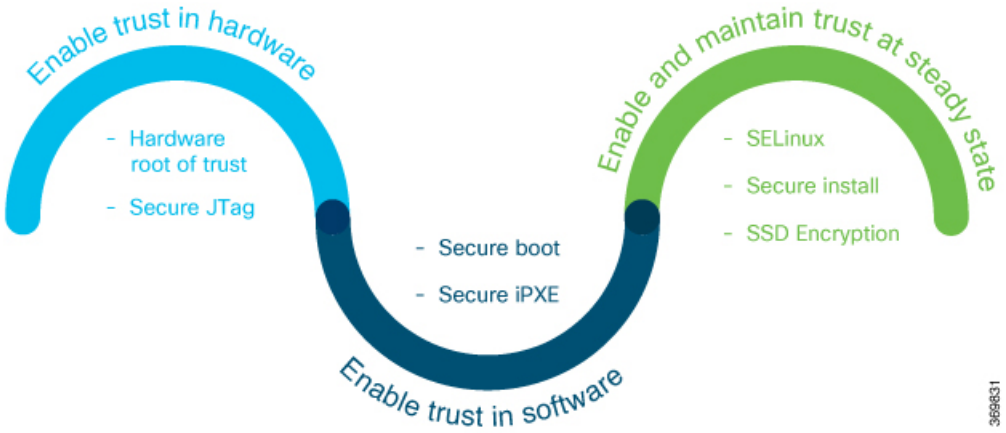
Need for trustworthy systems

Complex computing and communications networks are essential for the uninterrupted operation of global service providers, enterprises, and government networks. The integrity of data and IT infrastructure is fundamental to ensuring network security and user trust. As personal data becomes accessible anywhere, anytime, users anticipate consistent access and security across all networks. With adversaries becoming more aggressive, the threat landscape itself is undergoing a transformation. It becomes even more crucial to protect networks from attacks by malicious actors and from counterfeit or tampered products.

Main components of trustworthy systems

The main components of implementing a trustworthy system are:

- Enabling trust in hardware with Hardware root-of-trust and secure JTAG
- Enabling trust in software with secure boot and secure iPXE
- Enabling and maintaining trust at steady state with Security-Enhanced Linux (SELinux), Secure install, and SSD Encryption



Cisco Secure Unique Device Identifier (SUDI)

Cisco's Secure Unique Device Identifier (SUDI) is an X.509 certificate, with its private key securely stored in the Trust Anchor Module during manufacturing, providing a tamperproof device ID. It plays a vital role in verifying the device's initial identity during onboarding and, more significantly, in confirming hardware authenticity and protecting against counterfeiting.

To uniquely identify a device, the SUDI combines the product ID and device serial number. As an IEEE 802.1AR compliant X.509v3 certificate (which can be RSA or ECC), it effectively prevents the cloning or spoofing of identity information, significantly aiding in the avoidance of counterfeit switches in the supply chain.

Trust Anchor Module (TAm)

The Cisco Trust Anchor module (TAm), a tamper-resistant chip embedded in Cisco 9000 Series Smart Switches , contains nonvolatile secure storage and acts as the foundation of the trust chain. Crucially, the manufacturer's public key is securely stored within the TAm chip of every Cisco 9000 Series Smart Switches .

Cisco 9000 Series Smart Switches incorporate ACT and ACT2 hardware chips to implement Anti-Counterfeit Technology (ACT) within the equipment.

The TAm includes the following things:

Identity

During the manufacturing process, a product's ACT2 chip is loaded with the Cisco Secure Unique Device Identity (SUDI) in the form of an X.509v3 ECDSA or RSA certificate (or both), along with its corresponding keypairs and certificate chains. This SUDI serves as the foundation for Cisco's hardware anti-counterfeit check and is also utilized for establishing the initial network identity.

Entropy

The ACT2 features a NIST SP 800-90B compliant entropy source, making it ideal for seeding host-based pseudo-random number generators.

Key Management

The ACT2 is capable of generating symmetric keys, as well as ECC and RSA asymmetric keypairs. Crucially, the symmetric keys and the private portions of these keypairs are never released from the chip. Access to these protected keys is facilitated solely through cryptographic APIs, and certificates can be enrolled for the keypairs generated by the ACT2.

Secure Storage

The ACT2 offers approximately 50 KB of host data storage, secured against physical tampering. This makes it an ideal repository for sensitive information like licenses and confidential data such such as credentials. Crucially, important switch keys and passwords are also safeguarded within this secure storage.

The TAm and SUDI in Cisco 9000 Series Smart Switches function similarly to a mobile device's IMEI (International Mobile Equipment Identity) number. Both serve as unique hardware identifiers while also providing additional functionalities.

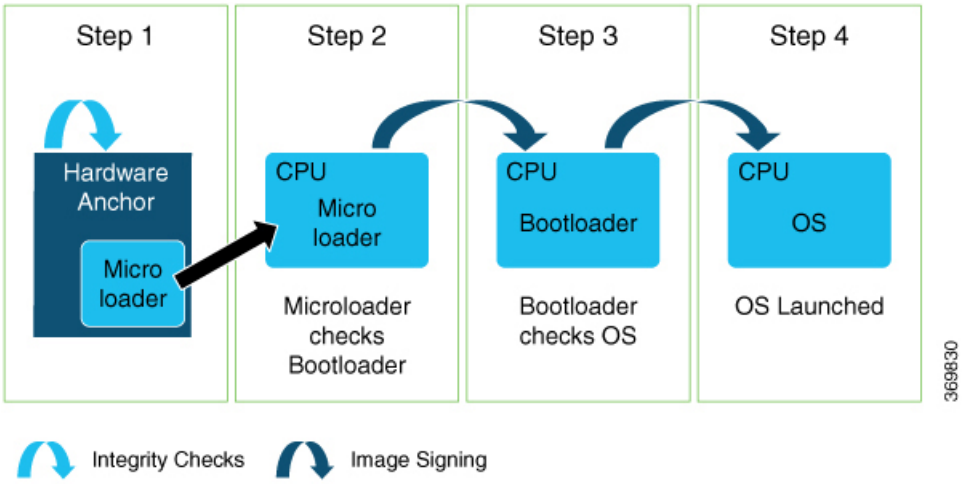
The TAm features secure nonvolatile storage for cryptographic keys, which applications can access via TAm libraries. During switch bootup, the SUDI and boot measurements stored here are accessed to verify the switch's legitimacy and ensure the booting process executes as expected, thereby detecting any tampering.

Secure Boot

To ensure the authenticity and integrity of code executing on Cisco 9000 Series Smart Switches , Cisco employs Secure Boot. Its hardware-anchored secure boot safeguards the micro loader—the very first piece of code that boots—within tamper-resistant hardware. This creates a root of trust, effectively preventing Cisco network devices from running tainted network software.

All images released by Cisco are digitally signed using the manufacturer's private key. This process occurs in an isolated, secure, and audited environment. The key pair itself is generated using RSA encryption with a 2048-bit modulus.

The device, after booting, authenticates with the TAm through SUDI to ascertain that its hardware is indeed genuine Cisco, rather than a counterfeit or compromised unit.



The public key, embedded directly into the hardware, is accessible via TAm libraries. Cisco IOS XE images are signed using private keys that are never removed from the build DevOps release environment. During the boot process, these keys are compared, confirming that both the hardware and software have been signed by Cisco.

The secure chain begins with the TAm, which first verifies the integrity of the microloader before it's loaded onto the CPU. In a similar fashion, the microloader then verifies the BIOS/bootloader by calling the TAm's API to ensure its image is legitimate before loading it. This same verification process is applied when loading the Cisco IOS XE image.

After the CPU loads the image, the operating system utilizes the TAm API to verify software modules like KLM, RPM, and ASIC SDK prior to their activation.

Chip Guard (Chip Protection)

Cisco 9000 Series Smart Switches incorporate a chip protection feature that guarantees hardware integrity. This protection mechanism records each device's signature during manufacturing and then compares it during every device bootup, thereby ensuring that the switch's peripherals are not counterfeit.

Manufacturing Time Database

The manufacturing time database is the original copy of the unique IDs of Cisco ASICs, CPUs, SoCs, and other devices with their device types specific to a board. In most cases, the unique ID is a device serial number or other appropriate value of that device. The manufacturing database is a Known Good Values (KGV) database specific to a board. It is programmed onto the TAm device as part of the manufacturing process.

Collected Database

The collected database is collected by the firmware whenever the board is booted and extended to the TAm device. Measurements are collected either through firmware or through system drivers.

The BIOS boot process integrates the TAm library to populate the collection database. The BIOS detects various hardware components as part of initialization and uses the TAm library APIs to record the device type and unique IDs if the detected devices are part of the manufacturing time database. After all the device types and unique IDs are written to the collected database, the platform operating system invokes the TAm library API to validate the collected database against the manufacturing time database. If there is a mismatch, the platform holds the boot process.

Random Number Generation and Entropy Source

Strong random number generation (RNG) is at the core of encryption, while weak RNG can undermine the entire encryption system. Random number generators play a key role in creating cryptographic keys, establishing highly secure communications between users and websites, and in resetting passwords for email accounts. Without assured randomness, an attacker can predict what the system will generate and undermine the algorithm. Cisco 9000 Series Smart Switches uses the RNG from Linux. The RNG is seeded with a random value, typically obtained from a hardware random number generator (HRNG), which makes it impossible to guess. Hardware also contains a Trust Anchor module that is compliant with NIST specifications and capable of providing much more effective RNG that extracts entropy from a true random source within the Trust Anchor.

Multistage BIOS

The BIOS is split into multiple, smaller pieces so that they can be loaded, validated, and executed entirely in the RAM to protect the BIOS from external modification (such as a Time-of-Check to Time-of-Use attack [1]). The BIOS is composed of the following things:

- Pre-EFI Initialization (PEI)
- Firmware Dependency Module (FDM)
- Driver Execution environment (DXE)

Having a multistage BIOS in Cisco 9000 Series Smart Switches makes it very hard to bypass the Cisco BIOS. Any intervention with the BIOS will stop the bootloader from loading the OS image

Runtime Defenses (RTD)

Runtime defenses target injection attacks of malicious code into running software. Cisco 9000 Series Smart Switches runtime defenses include Address Space Layout Randomization (ASLR), Built-in Object Size Checking (BOSC), and X-space Runtime defenses. These defenses make it harder or impossible for attackers to exploit vulnerabilities in running software.

Address Space Layout Randomization (ASLR)

Address Space Layout Randomization (ASLR) is an important security hardening functionality that randomizes the locations of sections of all processes and the kernel for Cisco 9000 Series Smart Switches to make it more difficult for an attacker to exploit existing vulnerabilities. ASLR is a companion defense along with executable space protection, which prevents inadvertent execution of code from unauthorized areas and prohibits writing of code over executable areas.

ASLR functionality for processes can be categorized into Cisco binaries and 3rd party binaries, both of which need to support ASLR. For ASLR support, Cisco and 3rd party binaries and shared libraries need to be built with the correct flags. Cisco binaries including 3rd party shared objects must ensure the library is randomized so as not to compromise the randomization of the Cisco binary itself. 3rd party binaries and shared libraries might require vendor support to randomize them.

ASLR functionality for the Linux kernel brings support for address space randomization to running Linux kernel images by randomizing where the kernel code is placed at boot time. Kernel ASLR support is present in Cisco 9000 Series Smart Switches , making it hard for the hackers to perform malicious code injections.

[1] A Time-of-Check to Time-of-Use (TOC/TOU) attack is a type of race condition vulnerability that occurs when a program checks a condition and then performs an action based on the result of that check. However, if another program modifies the condition between the time of the check and the time of the use, the action will be performed based on the modified condition, which could lead to unintended consequences.

Executable Space Protection (XSpace)

Executable space protection (X-space) is one of the most important security protections in Cisco 9000 Series Smart Switches . This feature ensures that executable space protection is enabled for Cisco devices, which prevents the execution of code from unauthorized areas and prohibits writing code over executable areas.

X-space makes Cisco devices more robust at preventing hackers from penetrating into the switches.

Object Size Checking (OSC)

Buffer overflow is probably the best-known form of software security vulnerability. A buffer overflow condition exists when a program attempts to put more data in a buffer than the buffer can hold, or when a program attempts to put data in a memory area past a buffer. In this case, a buffer is a sequential section of memory allocated to contain anything from a character string to an array of integers. Writing outside the bounds of a block of allocated memory can corrupt data, crash the program, or cause the execution of malicious code. switches have full protection to determine buffer overflows in C or C++ code by having object size checks before a write call.

SafeC Libraries

Cisco IOS XE software uses efficient library functions that promote safer, more secure C or C++ language programming and are based on the ISO/IEC 9989:2011 (C11) specification. Several standard C library functions are susceptible to vulnerabilities that can serve as launch points for more sophisticated attacks.

While providing "safe" replacements for standard functions in a consistent naming schema, SafeC aims to mitigate security exploits due to buffer overflows, provides bound checks that may not be present in the native library, and prevents string termination and truncation errors. This SafeC safeguards Cisco 9000 Series Smart Switches hardware from buffer overflow attacks.

Cisco Signed Kernel Modules

Signed Kernel Modules ensure that any kernel modules loaded into the system are authentic and unmodified. This prevents unapproved and untrusted executable code from being loaded into the kernel by conventional means.

If the modules are not signed by Cisco, then they cannot be used with the Cisco IOS XE. This feature stops unauthorized software from running on Cisco devices. All these hardening features make the Cisco IOS XE and its peripheral components difficult to hack.

Secure JTAG (sJTAG)

JTAG was also adopted to program FPGAs and provide a CPU debug access port. A laptop and a JTAG debugger are often all that is required to provide access to an embedded CPU allowing for retrieval of firmware images, dumping memory, and monitoring software execution. A small size interface coupled with a sophisticated toolset gives attackers a portable yet powerful means to exploit a system.

By having a secure JTAG on Cisco 9000 Series Smart Switches , we can mitigate intellectual property (IP) theft and avoid the stealing of passwords or keys from the memory.

Secure Erase

The secure erase feature erases all customer information within Cisco 9000 Series Smart Switches . Secure erase is an operation that removes all the identifiable customer information in Cisco IOS XE devices for purposes of product removal due to Return Material Authorization (RMA), upgrade or replacement, or system end-of-life.

- RMA for a device: If you must return a device to Cisco for RMA, remove all customer-specific data before obtaining an RMA certificate for the device.
- Recovering a compromised device: If the key material or credentials that are stored on a device are compromised, reset the device to the factory configuration, and then reconfigure the device.