KEYFACTOR

EBOOK

Eight Steps to IoT Security

A practical guide to implementing security in IoT devices

Table of contents

About the author:

Guillaume Crinon

Director of IoT Business Strategy, Keyfactor

Guillaume has 28 years of experience in the semiconductor and software industry, IoT security, radio-frequency circuit design, project and team management, and business-development. For the past 8 years he has been focusing on IoT security technologies and the way to bridge the security gap between IT / enterprise platforms and low-power connected devices.

Guillaume joined Keyfactor as Director of IoT Business Strategy with the global IoT team in 2022.

Before transitioning to this role, he was the IoT Strategy Manager for security and connectivity with the AVNET Global IoT team in charge of Avnet-specific products, such as a family of reprogrammable SIM cards (eUICC) and leading the development of a management platform focusing on delivering secure over-the-air provisioning and upgrade services to IoT devices.

Guillaume graduated from SUPELEC in Paris (MSc in EE) in 1994 and has co-authored 13 international patents in wireless systems and IC architectures/security to date.

Introduction

Implementing cybersecurity for IoT devices doesn't have to be complex.

Complexity occurs when product teams treat security as an afterthought in the global IoT project, rather than considering and implementing security at the design stage. Sometimes this is due to a lack of expertise by the product team. Other times a new project must leverage older hardware platforms and manufacturing processes that weren't designed with security in mind.

Device manufacturers must grasp three important concepts in cybersecurity:

1 Security is a journey, not a destination

There is no such thing as "perfect security." Standards will evolve. New tactics and requirements will be put into place. Security must be designed to evolve as well. It is better to have a measured approach to security and implement a plan in phases than to try unsuccessfully to implement everything all at once.

2 There is no "one size fits all" security plan

Every project will have restrictions. Whether it's power consumption, processor power, or bill of material and budgetary restraints, each project will be different. Therefore, each cybersecurity plan must also be tailored.

3 On new projects, security needs to be thought of at the beginning, not tossed on at the end

Early planning is essential for hardware and chip selection, as there are numerous factors to consider. It is advisable to prioritize security rather than compromising it to accommodate an earlier decision.

You must design hardware that will be flexible enough to support security not just for today, but also for the future. This means anticipating and monitoring the emergence of not-so-distant standards, regulations, and technology. Your product may not implement the foreseen security all at once, but the hardware should be designed to accommodate the targeted security upgrades to come and be capable of handling future security requirements.

With the appropriate hardware platform, a well-designed process, and a phased approach to implementation, you can constantly improve your product security by means of secure firmware upgrades throughout the product's useful life.

If you are brand new to PKI, here are some frequently used terms to know:

Learn more ↗

Eight steps to IoT security

This eight-step framework will help you plan for security as you start your next project. Many of these steps are interconnected and the order you execute them may change based on your organization's level of cybersecurity maturity.

Start your journey below ↓

STEP 1

Identify the standards and regulations your product needs to comply with, now and in the future

EXPERT INSIGHT

If your product is not governed by any standard, consider standards employed by adjacent industries with similar characteristics to identify security best practices.

Standards will vary based on your industry and geography, but they are the foundation for your device. Whether they are government-mandated or suggestions for interoperability by an industry consortium, standards provide value by serving as a blueprint for what is expected of your product. Standards will also go through iterations, so it is imperative to keep a pulse on how the standards are evolving.

Your final product should be standards-compliant or, if the standard is not complete, be capable of accepting an update to make it compliant. To define compliance requirements is to define your product's end state, so you can then work backwards to design it.

EXAMPLES

Industry standards

Automotive

ETSI C-ITS

ISO 21434

UNECE 155/156

WP29

ISO 15118-2

ISO 15118-20

Consumer

Matter

ETSI EN 303 645

Industrial

RFC 8995 (BRSKI)

IEC62443

Metering

WiSUN

EXAMPLES

Regional regulations

EU NIS2 directive

Strengthening American Cybersecurity Act

EU Cyber Resilience Act

GDPR

LEARN MORE

Matter — a cybersecurity perspective

Discover the ins and outs of the Matter standard and how manufacturers can implement PKI to improve the security of connected devices.

Download eBook ↗

STEP 2

Choose your device Root of Trust: hardware or software

EXPERT INSIGHT

Good security depends on a vault where the device's most sensitive secrets can be kept safely, made available when needed, and is incapable of being overwritten by malware or accessed by unauthorized users or systems.

A Root of Trust (RoT), or device trust anchor, is a secure element, trusted platform module (TPM), or microcontroller where device secrets are stored, and low-level cryptographic algorithms (primitives) can be safely executed. The RoT can take the form of either secured hardware or secure software on non-secured hardware. Deciding which to use will depend on Bill of Materials (BOM) cost, security requirements, and whether you have the physical space available for multiple chips.

Many types of secrets may need this extra protection. The number of secrets and processes will steer the decision on which RoT to use.

In addition to securely hosting secrets, an RoT can also protect the execution of cryptographic primitive functions or algorithms. It is possible for inadvertent "leaks" of secrets by a device if there is an improper software implementation, memory management, instantaneous power consumption glitches, or electromagnetic emissions, so an RoT is imperative.

Examples of secrets:

- Initial device certificate and private key pair that attest the genuineness of the device
- Software/firmware signature verification keys and/or certificates
- Symmetric session keys derived from TLS or equivalent security protocols
- Certificates that the device should trust when interacting with other machines
- Operational certificates and other unique private key pairs which the device will use during regular operations to secure end-to-end connections with other devices or servers

To combat these leaks, silicon suppliers offer many options:

- Secure elements, which may come with strong certifications such as Common Criteria
- Secure microcontroller units (MCUs) used in sensitive applications such as point-of-sale terminals used for payments
- Ordinary MCUs with Trusted Execution Environments that provide isolation between stacks, processes, and memory
- Secure software stacks running on the ordinary MCUs, which can obfuscate secrets and prevent leaks

The choice of the right RoT depends on the product design objectives:

- Number of secrets to be protected
- Format and size of these secrets (required space, memory, and processing power)
- Algorithms supported: RSA, ECC, etc.
- Capability to update secrets and/or cryptographic algorithms once the device is deployed into the field (i.e., crypto-agility)
- BOM cost for the device
- Business model of the device producer or service provider
- Expected lifetime of the IoT device and service provider warranty requirements

The RoT is immutable — it's burned into the hardware so that it can't be updated without removing it completely and starting over with new hardware. Consequently, the RoT cannot be upgraded with over-the-air (OTA) updates once the devices are in the field.

EXPERT INSIGHT

The choice for the root of trust must be made with a long-term vision of the project.

STEP 3

Selecting an MCU with enough memory for secure OTA updates

Your device software will evolve and grow over time, but predicting the precise future hardware requirements can be a moving target. Nevertheless, you will need to assess the most probable Flash and RAM size numbers that your device will require in the future, with consideration for the operating system and various stacks in addition to your own application. Additionally, you may want provision memory to enable updating your MCU's peripherals like a BLE or Wi-Fi chipset.

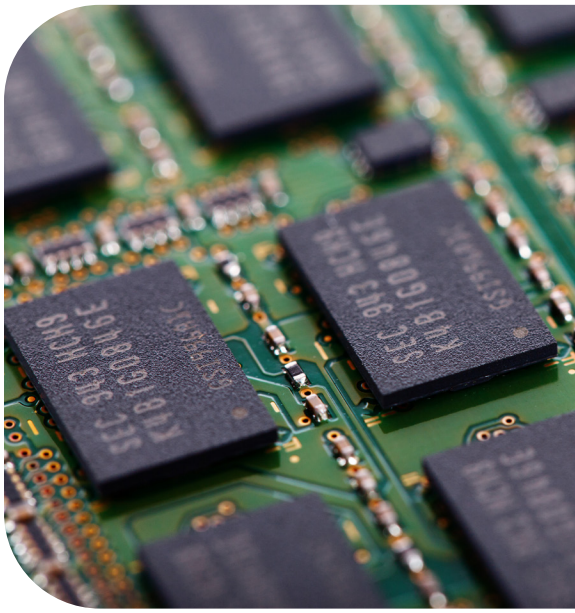
You will need to find the balance to make sure that you will not fall short of memory halfway through your product lifetime or warranty period that you commit to your end customers — while avoiding the costs of memory you will not use.

It is highly recommended to choose a secure OTA firmware update mechanism and implement its embedded agent. It will add value to the device at no cost; the best OTA technology providers offer their agent in source code for free. Your customer will then have the option to subscribe to the service you are prescribing or choose another one.

In a similar way, it is recommended to lay the groundwork for securely administering the Root-of-Trust and Trust Store, which your customer will need to do when operating the device. Like OTA firmware updates, some technology providers offer their agent in source code for free. Your customer will again have the choice to opt in or manage it their own way.

EXPERT INSIGHT

If firmware is designed to accept small patches or incremental delta updates, the amount of extra memory can be decreased. However, it may be simpler to push full images and just use double the memory.



STEP 4

Set up your Public Key Infrastructure (PKI) for initial device certificates

EXPERT INSIGHT

It is imperative to understand that your product's PKI and root CA are the foundation of the product's security. They should never be shared with anyone else.

As a manufacturer, your product is physically branded with a sign like your name, your trademark, or your logo which will make it easily recognizable in the physical world.

Similarly, your product will most likely communicate and interact with other machines in the digital world. Therefore, it needs a digital "sign" or document to identify and authenticate itself to prove its genuineness whenever needed. This is known as an initial device certificate.

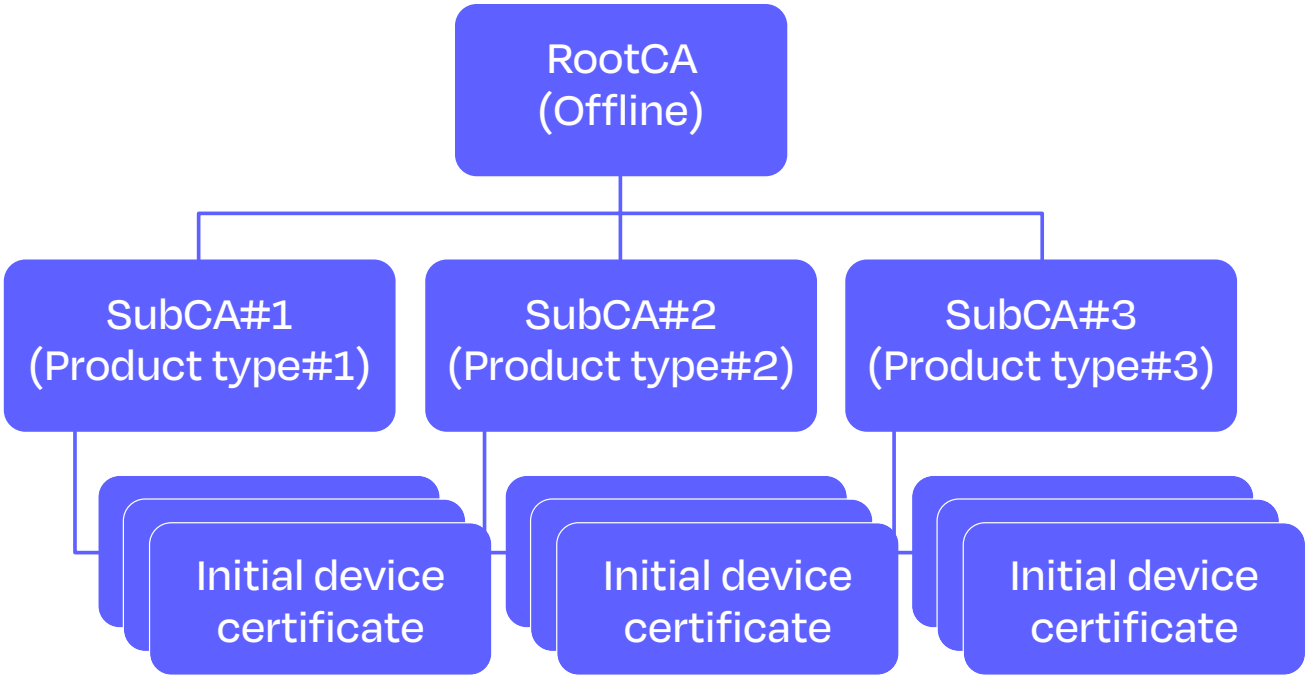
To issue these certificates, you need to set up and configure a PKI. A PKI consists of a root Certificate Authority (CA), which is safely kept offline, and a subordinate CA which is dedicated to each product line. The subordinate CA generates exactly one initial device certificate for every device manufactured. The initial device certificate is mathematically tied to the physical device's RoT and can be cryptographically traced back to the root CA. It proves the authenticity of the device in any digital transaction.

The lifespan of the initial device certificate is usually aligned with the expected lifetime of the device. However, just like never changing your password creates a security vulnerability, so, too, does a long certificate lifespan. To compensate for this, the certificate should be stored securely inside the device RoT and with strict limitations on its usage.

You should also consider where to host your PKI. Today's market offers multiple options based on your preferences and available resources — through a SaaS model with a vendor, hosted at your own facilities, either on-premise or in your cloud, or a hybrid model with both local and cloud-based components.

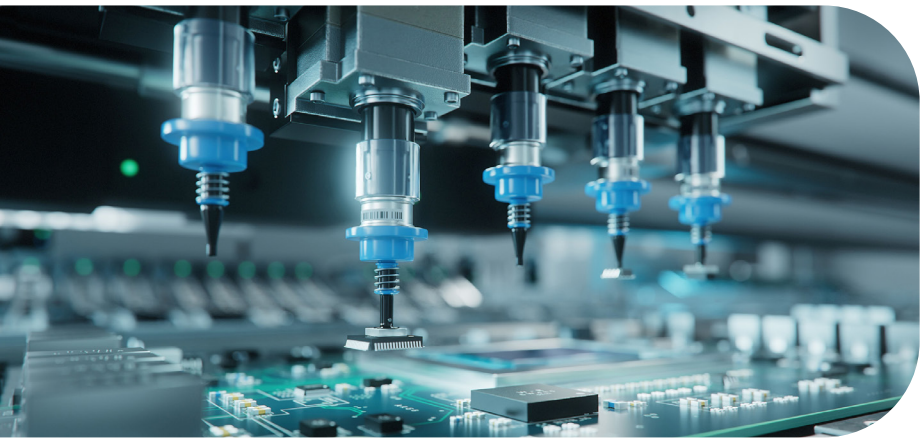
Although some silicon vendors offer easy personalization configurations where the root CA is owned and operated by them and shared among their customers, it is recommended that you own and run your own PKI within your company's or enterprise's control – remember, it's your product and your brand on the line, not the silicon vendor's.

> "The use of PKI is growing, and more machines use certificates to encrypt communication for authentication or to sign workloads (i.e., code signing.)"
>
> Managing Machine Identities, Secrets, Keys and Certificates, Erik Wahlstrom
>
> Gartner, 16 Mar 2022



STEP 5

Decide how initial device certificates will be delivered and injected into your production line



Now that your PKI is ready to issue initial device certificates, you must decide how the certificates will be put on your devices.

Depending on the preferences, processes, and capabilities of your company, there are three primary options for provisioning identities:

1. Pre-provisioned before manufacturing by your silicon vendor or distributor
2. Provisioned during manufacturing at the factory
3. Post-provisioned over the air after manufacturing

EXPERT INSIGHT

This decision may influence or be influenced by purchasing processes with silicon vendors, selection of manufacturing partners, security capabilities within your factory, and end-user experience.

OPTION 1

Pre-provisioned before manufacturing by your silicon vendor or distributor

Most silicon vendors and distributors offer this personalization service for secure elements. Less frequently, they may also offer it for MCUs. This option is convenient when you know which chip is going into which device. It also minimizes the PKI infrastructure required in your factories, which can be problematic if you have concerns about security or space limitations at your facility.

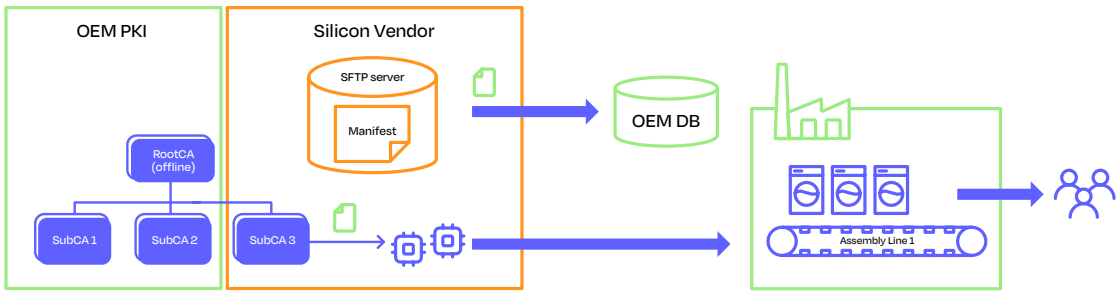
However, if you utilize the same hardware platform for several families of products, this option will become difficult to manage due to each product line requiring different certificate authorities.

Further considerations for this method include:

- You become reliant on an outside source for device identities
- Additional manufacturing and security processes with your silicon vendor
- Security processes are outside of your control

IoT identity provisioning happens at the silicon vendor factory

Before device manufacturing in the OEM factory



OPTION 2

Provisioned during manufacturing at the factory

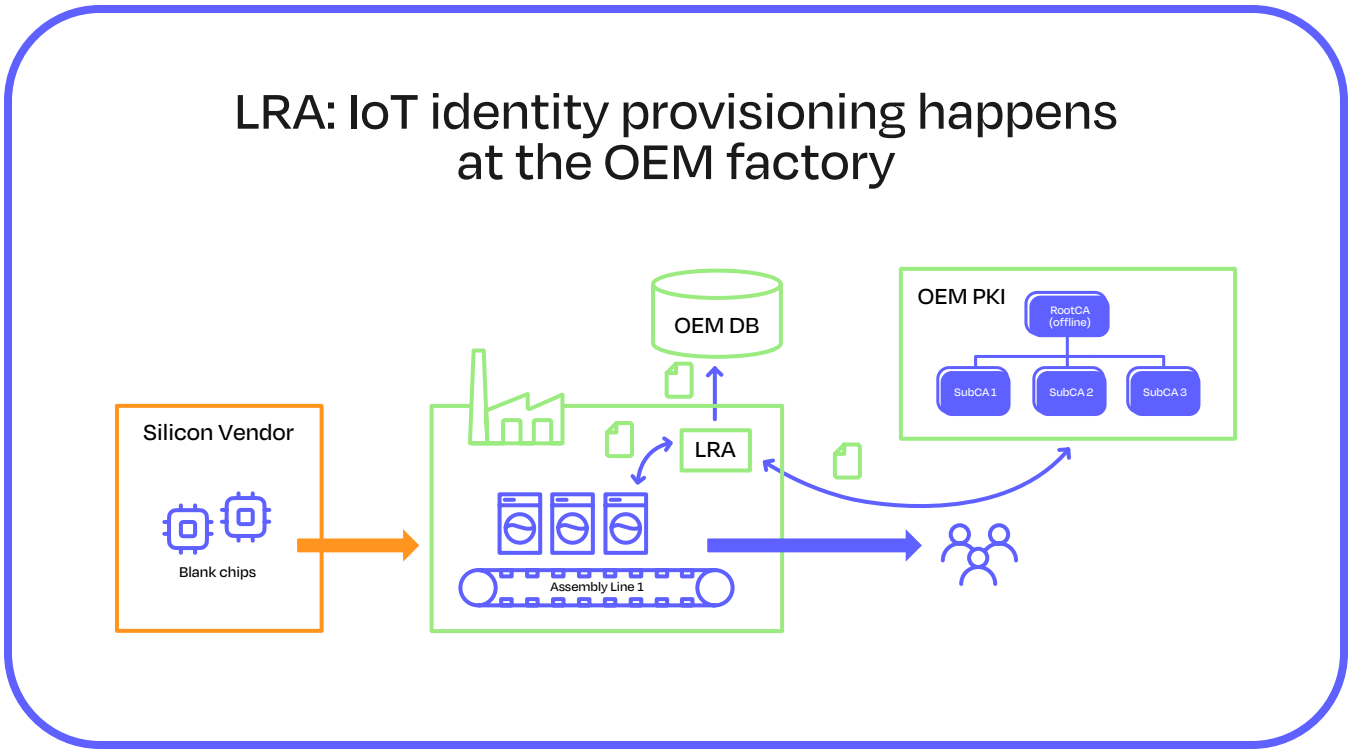
This option is the most flexible for a manufacturer, as you maintain complete control of your PKI and do not rely on any outside vendors, but it also is the most intensive as the manufacturer assumes all responsibility.

In this method:

- You must equip your factories with a local registration authority (LRA)
- The LRA collects certificate signing requests from devices being manufactured
- Then it sends the requests to your PKI and retrieves the certificates from the PKI
- It injects the certificates into the devices

EXPERT INSIGHT

If your PKI is hosted outside of your factory, this option only works if the factories can connect to the PKI in a reliable and secure manner.



OPTION 3

Post-provisioned over the air after manufacturing

In this scenario, devices are manufactured with chips that are pre-provisioned with a birth certificate from the silicon vendor PKI or a TPM. This provisional birth identity is then used by the device upon power-on reset to contact an enrollment management service. That service will vet the device's birth identity and, if cleared, collect an initial device certificate from your PKI. The initial device certificate will then be provisioned into the device, closing the manufacturing process.

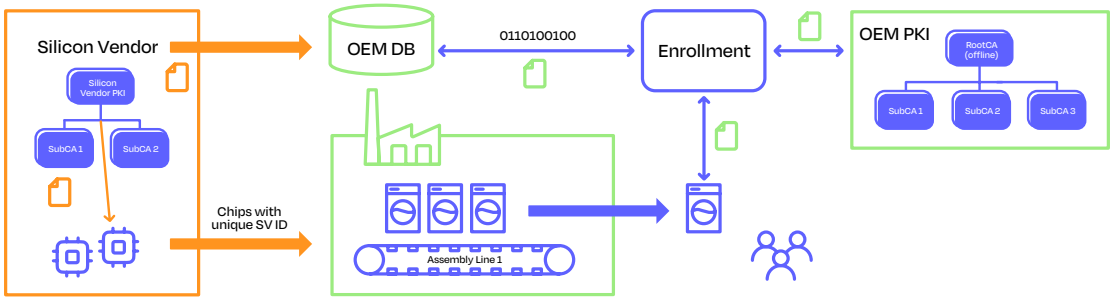
This option enables specific product design options, like the addition of region-specific certificates if you're producing devices for multiple countries from a single manufacturing location.

EXPERT INSIGHT

This option impacts end-user experience, as it requires internet access at the initial device onboarding for enrollment, which must be supplied by the user.

IoT identity provisioning happens Over-the-Air

After device manufacturing, before shipping to customer or in the field



STEP 6

Set up a role-based code signing procedure and create code signing keys

As an OEM, you have the responsibility of producing firmware and/or software for the device, but you must also maintain that code over the lifetime of the device. This software will be an assembly of:

- Operating system
- Various stacks for security, connectivity, etc.
- Software Design Kit (SDK)
- Application(s)

Regardless of how the code will be implemented and deployed, it needs to be cryptographically signed before being pushed to a device. This is necessary so that its authenticity can be checked before being locally applied. You can minimize the risk of malware or unverified software compromising the device by implementing the recommended code signing policies below.

EXPERT INSIGHT

It's important to note that MCUs may differ in the way they support code signing verification. Some make use of keys while others allow certificates. It is your responsibility to design a code signing process and to implement signature verification software according to what your MCU supports.

Recommended code signing policies:

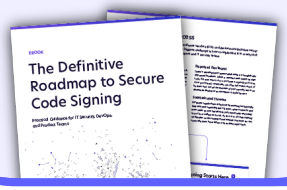
- Host your signing key securely on your code signing infrastructure
- Implement a signature verification algorithm in the device's root-of-trust
- Utilize different keys for production software, certification software, factory test software, etc.
- Adhere to strict processes to allow, control, and audit who signs your code releases
- Develop controls and auditing for who has access to the key and during what timeframe

EBOOK

The Definitive Roadmap to Secure Code Signing

Practical Guidance for IT Security, DevOps, and Product Teams

Download now ↗



Code signing is a powerful tool — if it's secure.

Signing code is a critical step in protecting your software supply chain, but it's more than just certificates and signatures. It's about ensuring only the right keys are used by the right developer to sign the right code and at the right time and place. Otherwise, it can open the door to malware, breaches, and supply chain attacks.

STEP 7

Implement a secure bootloader into your device

What if your secrets are well kept inside the device root-of-trust, but the boot software can be rewritten to bypass security controls? A bootloader prevents this from happening.

A bootloader is an immutable piece of code whose most important role is to execute at power-on reset and check the signature of the next executable routine using the device RoT. Like an RoT, the bootloader's immutability means the code cannot be changed, fixed, or updated — **ever.**

To minimize the opportunity for error, keep your bootloader small, simple, and limited to its primary function. To do this, the bootloader checks the integrity of the next stage of code to be executed, then:

- If code integrity passes, boot to the next stage
- If code integrity fails, take a prescribed action. This will vary based on your device and use case but possible options include:
  - Stop booting and request manual fix
  - Boot into failsafe mode and contact a quarantine service with recovery server
  - Boot to the faulty code and report the issue to a user and/or admin

EXPERT INSIGHT

This secure boot process can be segmented into several stages. Each stage initiates new features such as stacks and networking and checks the integrity of the upcoming stage until the final application runs.

STEP 8

At manufacturing, add the code signing verification keys and certificates to your device

Software verification keys and certificates are required onboard every device to check the authenticity of any software update before applying it. Unlike initial device certificates, verification keys and certificates are the exact same on every device manufactured.

Although these signature verification keys and certificates are public information, they should be hosted securely into a memory which cannot be overwritten, such as a protected memory or a secure element. Otherwise, an attacker could replace them with their own keys and push malware into the device, which would then be accepted, based on the attacker's keys.

EXPERT INSIGHT

These keys and certificates must be provisioned into every device at manufacturing. Since you've already selected a method for initial device certificates in step 5, you can utilize that service for this new material.

Conclusion

In our interconnected world, security is no longer optional. It is expected.

Bad actors are always looking for the weakest link in a network, which can be used to exploit the next device in the chain. As shown, cookie-cutter security cannot simply be added after the product design is finished. There is infrastructure, design considerations, along with processes and tools that must be put into place for a device to have proper cybersecurity.

However, you cannot just design for the needs of today. Your initial software won't be perfect. Bugs will be uncovered. Vulnerabilities will be discovered. Adjustments will be required to fulfill the device's operational specifications. Standards will evolve. But all these corrective actions can be taken with secure software updates to the connected device when security is designed in the device.

88%

of organizations agree improvements to IoT security are needed

2023 Navigating the State of IoT Security Report

Learn more

At Keyfactor, we're tracking the exploding number of machine identities driven by widespread IoT adoption.

Managing machine identities and securing devices will only become a more difficult challenge as organizations across all conceivable verticals find new use cases for connected devices.

From connected vehicles to pacemakers, we provide tools for all organizations to implement identity-first design methodologies. Our end-to-end platform for IoT helps streamline the entire PKI and certificate lifecycle to deliver quality, build trust, and scale with confidence.

No matter where you are in your security journey, our experts are here to help. Book a demo any time to tell us more about your organization and IoT goals.

KEYFACTOR

Keyfactor brings digital trust to the hyper-connected world with identity-first security for every machine and human. By simplifying PKI, automating certificate lifecycle management, and securing every device, workload, and thing, Keyfactor helps organizations move fast to establish digital trust at scale — and then maintain it. In a zero-trust world, every machine needs an identity and every identity must be managed. For more, visit keyfactor.com or follow @keyfactor.

Contact us

- www.keyfactor.com
- +1 216 785 2946 (North America)
- +46 8 735 61 01 (Europe)