Version 1: March 2022

Essential Properties of Secure Connected Devices

PSA Certified's 10 Security Goals and Microsoft's Seven Properties of Highly Secured Devices

Rob Smart, Senior Principal Security Architect, Arm (a PSA Certified Co-founder)
with contributions from Microsoft

This document was written by PSA Certified and Microsoft to illustrate the common objectives of the PSA Certified 10 Security Goals and the Microsoft's Seven Properties of Highly Secured Devices. It is well suited to people looking to understand, at a high-level, what is necessary for a connected device to be secure, based on expertise from two key industry players. Key things we will cover are:

- The perspectives and motivation of Microsoft and PSA Certified to identify the foundational security requirements.
- An overview of Microsoft's Seven Properties of Highly Secured Devices and the PSA Certified 10 Security Goals.
- How the seemingly different sets of properties and goals, in fact have a great deal in common.

psacertified™

Microsoft Azure

We enter the era of digital transformation in which nearly every industry is embracing technologies that make new efficiencies, products, and services possible at a scale never seen before. At the heart of this digital transformation is the ability to connect devices, collect and interpret data, and deliver intelligent business or service insights. However, insufficient investments in securing the devices that underpin the fundamental value for businesses have left both consumers and enterprises exposed to severe security risks.

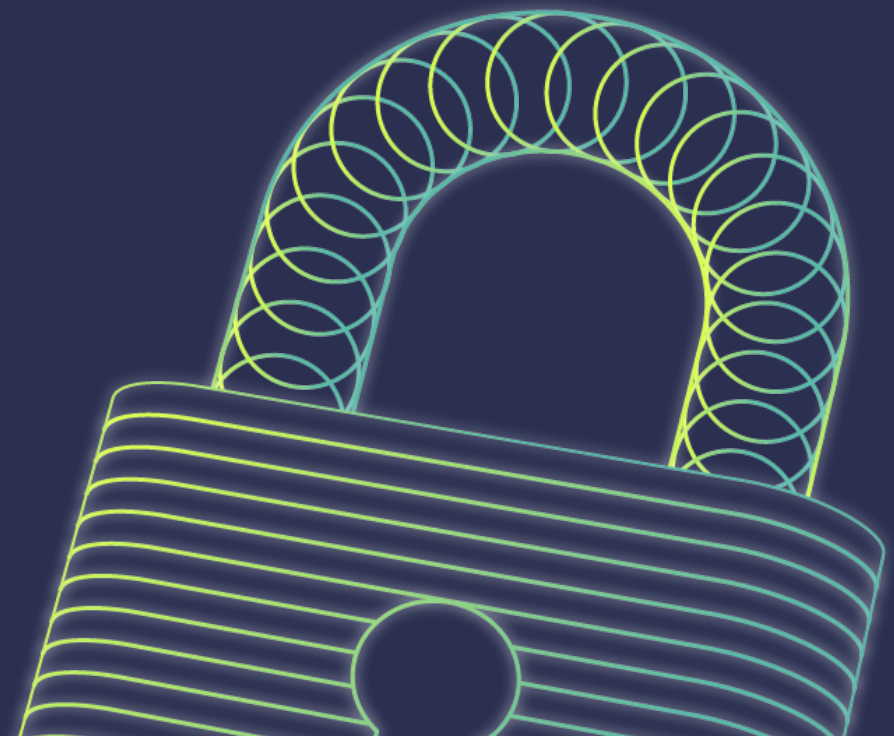
Ultimately, digital transformation will be powered by data: authentic, trustworthy data is more important than ever. The small quantities of data collected from every sensor and actuator must be trustworthy as it cumulatively becomes the big data driving the new transformational digital services at scale. Trusted data and trusted services can only be truly possible and achieve scale if they are generated by devices built with sound security principles.

As IoT started to grow, Arm and leading security evaluation laboratories spearheaded PSA Certified [1], a partnership that set out to establish baseline security requirements with the ecosystem. This was based on the PSA Certified 10 Security Goals that every connected device should meet before interacting with the Internet. These can be realized with the inclusion of a Root of Trust (RoT) and they motivated the definition of the PSA-RoT. The goals are a high-level abstraction derived from long-established Arm experience in securing devices, and from specific threat modeling and security analysis for common connected device use cases.

At the same time, Microsoft had observed that high development costs and maintenance often limited the adoption of strong security in the connected devices ecosystem. Every single device, be it a connected thermostat or equipment connected on a factory floor, is a potential target for an attack and therefore necessitates high-integrity security. Through extensive research and testing, Microsoft identified seven properties that should at a minimum be present in all devices considered to be highly secured. The results of that research is documented in The Seven Properties of Highly Secured Devices paper [2].

Both the PSA Certified 10 Security Goals and the Microsoft Seven Properties aim to advance the adoption of foundational security in the IoT device ecosystem. In this white paper, we provide a description of both with material from both PSA Certified and the Seven Properties of Highly Secured Device paper, to present a high-level comparison and our perspectives on their similarities and differences.

Ultimately, the call-to-action is simple: security is everyone's responsibility, and we need to rally together to advance the security of the IoT.



What are the Microsoft Seven Properties of Highly Secured Devices?

The following excerpts from The Seven Properties of Highly Secured Devices (2nd Edition), © 2020 Microsoft Corporation, and references to it used throughout the remainder of this paper, are used with permission. These excerpts are provided "as-is." The information and views expressed in these excerpts, including URL and other internet website references may change without notice. You bear the risk of using it. Some examples may be for illustration only and are fictitious. No real association is intended or inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

Microsoft conducted extensive research and testing to understand the baseline requirements for security in connected devices. The resulting evidence informed the paper "The Seven Properties of Highly Secured Devices." That paper details the seven properties found in every device considered to be highly secured, forming a foundation of security upon which additional security measures are often added. These seven properties should be considered a required baseline for security in every connected device. For any property that is missing, other practices would need to be implemented by the owner or customer to compensate. For example, a security incident might make it necessary to disconnect devices and recall or manually patch them without renewable security [3].

The Seven Properties of Highly Secured Devices

Is your device highly secured or does it just have some security features?

Dynamic Compartments
Can your device's security improve after deployment?

Hardware Root of Trust
Is your device's identity and software integrity secured by hardware?

Password-less Authentication
Does your device authenticate itself?

Defense in Depth
Does your device remain protected even if some security mechanism is defeated?

Error Reporting
Does your device report back errors to give you in-field awareness?

Small Trusted Computing Base
Is your device's security-enforcement code protected from bugs in application code?

Renewable Security
Does your device software update automatically?

aka.ms/7properties

Highly secured devices have a *hardware root of trust.*

A device's private identity keys are protected by hardware, the integrity of device software is validated by hardware, and the hardware contains physical countermeasures against side-channel attacks. Unlike software, hardware has two important properties needed as foundation for device security. First, single-purpose hardware is resistant to reuse by an attacker for unintended actions. Second, hardware can detect and mitigate against physical attacks; for example, pulse testing the reset pin to prevent glitching attacks is easily implemented in hardware. When used to protect secrets and integrity, hardware provides a solid root of trust upon which rich software functionality can be implemented securely and safely.

Highly secured devices have *defense in depth.*

In highly secured devices, multiple mitigations are applied to each class of threat. In devices with only a single layer of defense, such as most RTOS-based devices, even a single error in design or implementation is sufficient to lead to catastrophic compromise. Because new threats are often completely unanticipated, in practice, having multiple countermeasures often becomes the difference between a secured device and compromised device.

Highly secured devices have a *small trusted computing base.*

A trusted computing base (TCB) is "a small amount of software and hardware that security depends on and that we distinguish from a much larger amount of software that can misbehave without affecting security". Within a device, the TCB for different operations with may differ. For example, the TCB for securing data at rest may include the hardware root of trust, software for encryption and decryption, and software for sealing and unsealing crypto keys. On the other hand, the TCB for secure communication might also include an TLS implementation. The TCB for any operation should be kept as small as possible to minimize the surface that is exposed to attackers and to reduce the possibility that a bug or feature can be repurposed to circumvent security protections. The TCB code should be protected from non-critical device code to ensure its correct operation even if the other code is compromised. Less secured devices often have no isolated TCB - security code in these devices executes in the same compartment as the rest of the device code with the result that just one bug, anywhere in the device's code, can lead to a catastrophic full-system compromise.

Highly secured devices provide *dynamic compartments*.

In a computing device, cybersecurity compartments are hardware-enforced boundaries that prevent a breach or flaw in one software compartment from propagating to other software compartments of the device. Compartments introduce additional protection boundaries to create additional layers of defense in depth. Dynamic compartments allow the introduction of new boundaries, throughout a device's deployed lifetime, as required to improve security against escalating security threats.

Highly secured devices use *password-less authentication*.

Password-less authentication, such as certificates, are used to prove identities for mutual authentication when communicating with other local devices and with cloud services. A certificate or other password-less authentication token is a proof of identity and authorization that is signed with a secret private key and can be validated against a known public key. Unlike passwords or other authentication mechanisms that are based on shared secrets, password-less authentication mechanisms, backed by a hardware root of trust, can't be stolen, forged, or otherwise used to authenticate an impostor.

Highly secured devices have online *error reporting*.

When an error occurs on a secured device, an error report is collected automatically and sent to an error analysis system in a timely manner. In the best case, the error was caused by inadequate programming for an extremely rare sequence of events. In the worst case, the error was caused by an attacker probing the device for new attack vectors. Whichever the case, an error analysis system correlates error reports across an entire fleet of devices to allow automated diagnosis of errors. With a sufficiently large reporting base, even extremely rare error conditions can be diagnosed and corrected, and new attack vectors can be identified and isolated before they are widely exploited. Error reporting enables a global "immune system" across a fleet of highly secured devices. Without automated online error reporting, device manufacturers are left in the dark as to the device errors experienced in the field and may be caught off guard by emerging attacks.

Highly secured devices have *renewable security*.

A device with renewable security can update to a more secure state automatically, even after the device has been compromised. Renewable security is necessary because security threats evolve and escalate as attackers discover new attack vectors and create new attack methods and tools. To counter emerging threats, device security must be renewed regularly. In extreme cases, when compartments and layers of a device's software are compromised by zero-day exploits, lower layers must rebuild and renew the security of higher levels of the device. Remote attestation and rollback protections guarantee that, once renewed, a device cannot be reverted to a known vulnerable state. A device without renewable security is a crisis waiting to happen.

What are the PSA Certified 10 Security Goals?

The PSA Certified Security Goals, on the right, provide a high-level abstract way to think about the essential features that secure and establish trust in a connected device. Based on best security practice from across the industry, the set of goals is broadly applicable to different entities in the supply chain, from chip designers, software developers and device vendors through to cloud and network infrastructure providers. Abstraction allows these goals to be applied as required, for example, to an end user connected device, a hardware component, a software component, or a service. In describing the goals, the term device is used to represent any entity at any level that must be secure and trustworthy. It should be noted that though the focus is on local or internet connected devices, many aspects are relevant to securing non-connected devices.

The PSA Certified Platform Security Model outlines 10 goals:

1 **Unique identification:** To interact with a particular device, a unique identity should be assigned to the device and this identity should be attestable. This identity facilitates trusted interaction with the device for example, exchanging data and managing the device.

2 **Security lifecycle:** Devices should support security lifecycle that depends upon software versions, run-time status, hardware configuration, status of debug ports and the product lifecycle phase. Each security state of the security lifecycle should be attestable and may impact access to the device.

3 **Attestation:** Evidence of the device's properties, including the identity and security lifecycle state of the device should be provided through attestation. The device identification and attestation data should be part of a device verification process using a trusted third party.

4 **Secure boot:** To ensure only authorized software can be executed on a device, secure boot and secure loading processes are required. Unauthorized boot code should be detected and prevented. If the software cannot compromise the device, unauthorized software may be allowed.

5 **Secure update:** When providing security or feature updates to devices, only authentic and legitimate firmware should be updated on the device. Authentication, at the time of download, may be performed however, the execution of the update must be authorized via secure boot.

6 **Anti-rollback:** Preventing rollback to previous software versions is essential to ensure that previous versions of the code can't be reinstated. Rollback should be possible for recovery purposes only when authorized.

7 **Isolation:** Isolation aims to prevent one service from compromising other services. This is done by isolating trusted services from one another, from less trusted services and from un-trusted services.

8 **Interaction:** Devices should support interaction over isolation boundaries to enable the isolated services to be functional. The interfaces must not allow the system to be compromised. It may be required to keep the data confidential. Interaction should be considered both within the device and between the device and the outside world.

9 **Secure storage:** To prevent private data being cloned or revealed outside the trusted service or device, it must be uniquely bound to them. Confidentiality and integrity of private data is typically achieved using keys, which themselves need to be bound to the device and service.

10 **Cryptographic and trusted services:** A minimal set of trusted services and cryptographic operations should be implemented as the building blocks of a trusted device. These should support critical functions including security lifecycle, isolation, secure storage, attestation, secure boot, secure loading and binding of data.

The PSA Certified Security Goals form the basis of the Platform Security Model [4], an overarching document that defines in an architecture-agnostic way the important concepts and terminology. It also motivates other PSA Certified specifications, some of which, and all those associated with the independent PSA Certified evaluation program, do not depend on any Arm-technology based implementation. However, facilitating the construction of secure products is valuable, so some platform security documents are guides on how to meet the requirements using Arm technology. Additionally, there are PSA Functional APIs [5] for standard services such as cryptography, secure storage and device attestation aimed at encouraging the adoption of such standard services.

Relating the Seven Properties and the 10 Goals

The PSA Certified program has mapped its requirements to leading international sets of security guidelines. Identifying the relationship between any two sets proves not to be a straightforward task, though often it becomes clear there is much commonality. In this paper, we aim to illustrate that this is true also for the relationship between the Seven Properties and the 10 Goals. To explore the relationship between the two in more detail, we will discuss, in turn, each of the Seven Properties and what that means in terms of the 10 Goals. There is no significance to doing it this way, we could just as easily look at each of the 10 goals in turn.

Hardware Root of Trust

A hardware Root of Trust, termed *Immutable Root-of-Trust* in PSA Certified, encompasses the hardware and any fixed firmware and data required to establish and maintain the device trust. It is the immutability of such items that means it can be inherently trusted. Unlike software, the immutability of hardware makes it difficult to reuse by an attacker for unintended actions.

Included is the first stage of boot immediately after release-from-reset. This must be secured as it is the anchor that ensures the integrity through signature verification of the software intended to run on the device. In other words, ensuring that only authorized software can be executed ④⑥.

The hardware Root of Trust is also responsible for the secure storage of immutable data. Examples include device identities ①, any private keys to be able to prove (or attest) the identity of the device ③, a public key (perhaps from a manufacturer) used to ensure the integrity of the software ④, irreversible storage for rollback protection ⑥, and any secrets such as unique encryption keys to protect data at rest ⑨.

Additional hardware is often necessary to enforce the security of the device. Examples include controlling access based on lifecycle states such as provisioning, secured and debug ②; controlling access via isolation mechanisms, including write lock and/or read lock on required data fields ⑦. Cryptographic hardware may be provided makes use of the immutable Root of Trust secrets, and trustworthy sources of entropy for random number generation ⑩.

Depending on the risk applicable to the target product or any required certification scheme assurance level, the hardware Root of Trust implementation may be tamper-resistant and may include countermeasures against certain classes of side-channel attacks perhaps through the use of hardware voltage and clock glitch detectors.

Small Trusted Computing Base

Connected devices are hugely reliant on software so it is important to distinguish between the software (and hardware) that fulfils the trusted security functions and the rest of the system software.

The aim is to minimize the attack surface of security critical functionality; the less code there is, the better defined the interfaces (8), the lower the chance of there being a vulnerability or a feature that can be repurposed to circumvent security protections. It is unlikely that all the required security functionality can be concentrated in one small verifiable piece of functionality. Different operations within a device may, therefore, rely on different TCBs. Isolation (7) of those TCBs allows each to be tested and verified separately and the attack surface of each is reduced to the interactions across the interfaces (8).

A small TCB is an implementation objective, not directly reflected in a security goal. The PSA Certified Platform Security Model recommends that the immutable part of the Root of Trust is small, simple and verifiable; it cannot be changed once the chip has been manufactured. Further Root of Trust functionality must typically exist in software. Code within a TCB should be rigorously tested for potential compromises by skilled experts well versed in the latest tools and methods employed by attackers.

The ability of non-critical software to impact the operation of any security critical code relies on isolation of that TCB (7). Isolation can be used to partition the security critical system into a set of isolated TCBs with defined interfaces that perform distinct security functions. These may be separably verifiable. Such isolation, though, is not enough.

In minimal implementations, critical cryptographic key operations should be isolated in a small TCB. For example, all device private identity keys and access to them should be limited to the smallest possible subset of the device hardware and software. This is an aim of the PSA Root of Trust. In more comprehensive solutions, the TCB may be layered to protect access to persistent storage, to protect access to critical I/O resources, to detect and recover from compromises in code above the TCB, and to fail over to protected backup software in the case of catastrophic device compromise.

The extent of the protection may well be influenced by any market requirements, perhaps via an applicable certification scheme.

Dynamic Compartmentalization

Isolation (7) is the basis of partitioning (compartmentalization) of system software and access to critical system resources. From the security perspective, the objective is often taken to mean isolation of more trusted software from less trusted software. However, isolation is really an acknowledgment that all software can contain flaws that could be exploited to compromise the security of the device. In a broader sense, isolation aims to prevent one software component from compromising another, and hence compromise the device and service. Such compartmentalization introduces protection boundaries that create additional layers for defense in depth.

Three isolation boundary concepts are identified within the PSA Certified Platform Security Model, though deployment will depend on the threats applicable to the target market, and any applicable certification scheme. As a minimum, hardware enforced isolation of the security services from the rest of the system is required. This maps naturally to Arm TrustZone™, but PSA Certified expresses all requirements in a way that does not require Arm-based solutions. For example, use of multiple processors with hardware filtering is a valid approach, as in some instances, is the of use operating systems processes or independent virtual machines. Of course, these techniques can be deployed in any system and can add to the notion of defense in layers.

Compartmentalization that is dynamic allows the isolation boundaries to change with new secure in-field updates (5) aimed at security improvements to counter newly emerged security threats and attacks. Less secured devices, typically low-cost devices employing a real-time operating system (RTOS), have either no software compartments or only a small or fixed number that cannot be reconfigured after a device is deployed and thus have very limited ability to evolve to address new security threats. Of course, this requires a secure update mechanism (5) and a way to revoke previous versions (6). However, supporting device update may be impacted by resource constraints, for example, volatile memory to process the download, non-volatile memory for software and data storage, One-Time-Programmable memory, and performance.

Isolation is essential, but often interaction between isolated components is necessary if those isolated components are to serve a purpose in the overall system (8). The design of those interfaces must ensure that they cannot be used, or abused, in order to compromise the security through using the function in an unintended way or result in disclosure of any sensitive data owned by that function or any system reconfiguration controlled through that configuration. For example, abuse of an interface must not result in the leak of cryptographic keys, or the disablement of mechanisms that establish the security of the system. Additionally, PSA Certified recognizes that it may be necessary to ensure the confidentiality and integrity of any data exchanged over such interfaces. This is especially true if the interface is to the outside world, where the ability to intercept the data exchanges is much greater than within the system.

Password-less Authentication

Authentication is concerned with securely identifying something. That something may be the software to be executed (4), the integrity of any update downloaded (5), the security state of the device reflecting state of the hardware and software in the device through attestation (2)(3), or the device identity (1) for mutual authentication when communicating with other local devices and with cloud services (8).

Password-based techniques, or more generally those that rely on a shared "secret", are prone to the secret being revealed. This is especially true for authentication protocols that require the password or secret to be communicated over a channel that can easily be monitored. A certificate or other password-less authentication token is a proof of authenticity that is signed with a secret private key and can be validated against a known public key. Unlike passwords or other authentication mechanisms that are based on shared secrets, password-less authentication mechanisms, backed by a hardware Root of Trust, cannot be stolen, forged, or otherwise used to authenticate an impostor. PSA Certified recommends the use of techniques that do not use a shared secret, which typically means the use of certificates and asymmetric cryptography.

Beyond minimum implementations, such as those that require boot of authorized code (4)(6) and device identity (1)(3), certificates can attest to the identity of the software running on the device (2)(3). This, for example, allows a verification service to check if the device is running the latest software, which addresses all known security vulnerabilities, or requires a secure update (5) and confirmation that it has been applied before the service proceeding. In other words, a connecting service must be able to verify that the claimed device identity indeed refers to the device it claims to be. This may rely on an embedded private key which must be protected by the hardware Root of Trust to ensure that it can never be revealed; this is essential to ensure the device cannot be spoofed.

Renewable Security

Implementing all functionality entirely in a hardware Root of Trust, including design-time code in a ROM, is risky. Should a fault be found that proves impossible to work around in any renewable part of the design, the only course of action is a new cut of the silicon, which can be very costly. Clearly, that only addresses future production and not parts already manufactured. Complete reliance on a fixed hardware solution also means no flexibility to accommodate different requirements and no ability to add new Root of Trust services.

These factors are why the PSA Certified Platform Security Model defines a minimal trusted computing base for the hardware Root of Trust and identifies the need for security services implemented in software that is authenticated by the hardware Root of Trust. *"A device with renewable security can be updated to a more secure state ⑤, ideally automatically, even after the device has been compromised. Renewable security is necessary because security threats evolve and escalate as attackers discover new attack vectors and create new attack methods and tools. To counter emerging threats, device security must be renewed regularly. In extreme cases, when compartments and layers of a device's software are compromised by zero-day exploits, lower layers must rebuild and renew the security of higher levels of the device. Remote attestation ③ and rollback protections ⑥ guarantee that, once renewed, a device cannot be reverted to a validly signed but flawed state. A device without renewable security is a crisis waiting to happen."* [2]

Ideally, each layer of a device's software defenses should be independently updated without invalidating other layers, possibly even automatically. Importantly, any update must be authenticated, ultimately linked via a chain of trust to the hardware Root of Trust and be robust to ensure that the update itself does not compromise the device ⑤. *"Reliable implementations of renewable security provide robust mechanisms to automatically recover even from failed updates or extraneous conditions such as power or network failures during an update."* [2] This allows for those practical cases where it is necessary to securely revert to a previous authenticatable version in the event that an update is shown to have issues after delivery.

Defense in Depth

A single error in design or implementation may be sufficient to lead to catastrophic compromise if there is only a single layer of defense. New threats are often completely unanticipated so, in practice, multiple layers of defense often make the difference between a secure and a compromised device. Common to almost all defense in depth designs must be the philosophies of "assume breach" and of "zero trust". The designers must evaluate the effectiveness of each protection mechanism if an attacker has already compromised other parts of the device.

Defense in depth can take on many forms. For example, executing only authorized code ④, preventing rollback to older versions with known vulnerabilities code ⑥, which, of course, means renewable security through a secure update mechanism ⑤. Isolation of the various codes ⑦ is a really important defense in depth technique to prevent the exploit of a compromise, for example, to prevent a networking stack overwriting a device's firmware in Flash storage. Running critical tasks on physically isolated cores is a strong mechanism for providing such isolation and can be relatively easy to reason about. The use, though, of processor privileged execution models with access control supported by Memory Management Units is very common. The use of virtualization can add another layer in the defense of device firmware protection.

Such isolation really tackles interaction not intended by the designer. Intentional interaction must be robust ⑧, in other words, there is little point in providing isolation if the intentional interaction points result in, for example, data leakage.

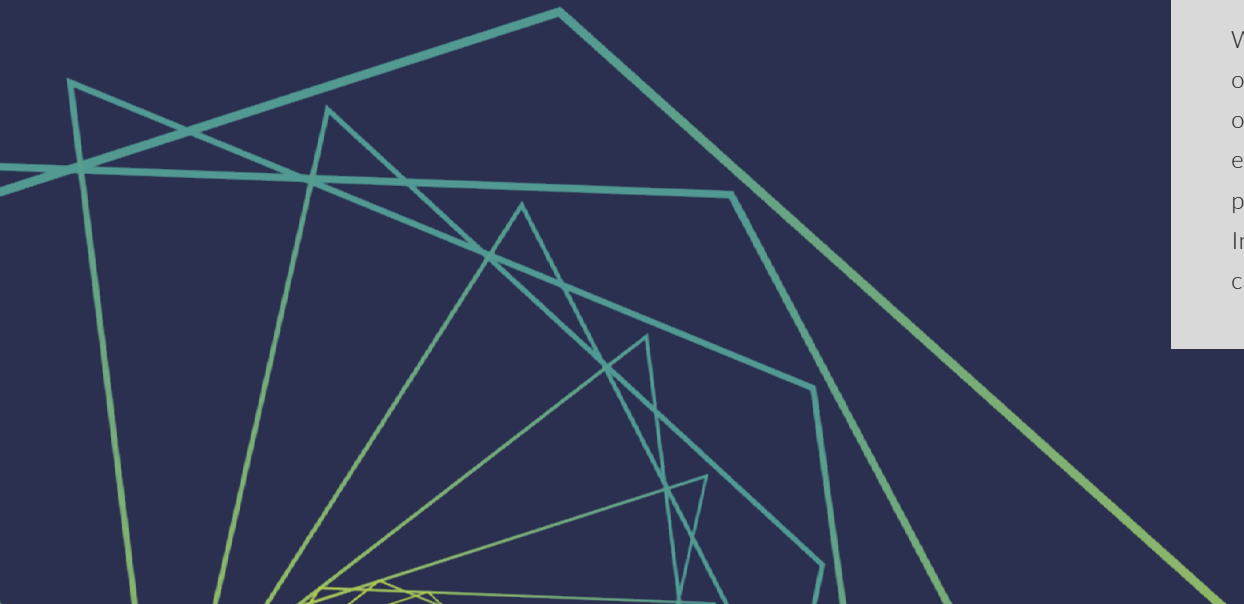
Hardware should also consider defense in depth. For example, write-once latches on configuration registers and write-protected latches on program code limit the extent to which a device can be repurposed even if the device's software is temporarily compromised. Such mechanisms are ways to implement the concepts touched on in the PSA Certified Platform Security Model through the partitioning of the RoT into immutable and mutable components, the notion of temporal isolation and the provision of trusted services to configure the security hardware and to support the security functions ⑩.

Multiple mitigations for identified threats is not expressed directly as a PSA Certified Security Goal. Beyond the PSA Security Goals, the PSA Certified specifications recommend hardware tamper resistance and countermeasures against side-channel attacks, access control to hardware resources, encryption of stored data, control of access to debug ports, and so on. These contribute to the depth of defense in a device. The extent of support will depend upon the target market, the cost implications and any protection profiles associated with certification schemes.

The Error Reporting Property

Timely reporting of errors detected on a secured device to an online error analysis system is one of the 7 properties of highly secured devices. Exactly which errors can be detected with sufficiency of information and reported, especially for the least predictable of issues, will be very implementation specific. Moreover, the reports must be analyzable by the online error analysis system and the broader eco-system must be able to determine the appropriate corrective action, neither of which is a function of the connected device. Error reporting is, therefore, outside the scope of PSA Certified and the goals. None-the-less, error reporting will lead to enhanced device security if the device is able to apply the corrective actions, for example, perform secure updates (5), prevent rollback to vulnerable versions (6), and allow a service to act based on the known state of the device (2)(3).

There are PSA Certified Level 1 requirements to log security related events and to secure the confidentiality and integrity of the logs [4]. The authenticity of any failure reports generated on a device can bound to that device following the binding goal (9), or via the attestation goal (3) and bound to the device by the PSA Certified Initial Attestation service.



The PSA Certified Security Lifecycle Goal

One PSA Certified goal that is not directly covered by the Seven Properties is that devices support a security lifecycle state. Each security state in the security lifecycle defines the security properties of the device, and will depend on things like the software versions, run-time measurements of what is on the device, the hardware configuration, the debug mode, and on the product lifecycle phase (for example, development, deployment, returns and end-of-life). This goal is motivated by the need for a connected service to act in a way that depends on the security state.

While not distinguishing between different lifecycle states, the Seven Properties of Highly Secured Devices provide a viewpoint on attestation. Using a certificate or authentication token that is issued by the device based on attestation, for example, enable services that are accessed by a device to verify and require that the device perform a software update and is not running an older version of its software. In addition, remote attestation and rollback protections guarantee that a device cannot be reverted to a known vulnerable state.

Conclusion

As the reliance on digital transformation, trusted data and connected devices rapidly soars, it's clear that businesses need to take the security of devices seriously. In order to protect the reputation of businesses and services, leaders need to be taking active steps to ensure that they are not introducing weak links into their systems via insecure devices. However, developing, manufacturing, deploying, and ultimately managing IoT devices securely can pose unique challenges.

PSA Certified's 10 Security Goals and Microsoft's Seven Properties of Highly Secured Devices offer informative checklists to work with. They demystify baseline device security requirements and help business leaders to understand what they should be requesting from suppliers, and aiming for in their own products. Implementing the necessary features to provide customers with confidence to trust their devices includes silicon requirements, such as a hardware-based root of trust, software architecture decisions, and deep integration with cloud services [3]. Due to this scope and complexity, there can be differences between the Seven Properties and the 10 Goals in how those requirements are implemented, and where the responsibilities fall to ensure they are met and managed. To help, the PSA Certified program includes threat models and security analyses, security requirement specifications and application programming interfaces, all of which are architecture-agnostic, together with an open-source reference implementation and test suites. Together, these enable consistent design-in of a Root of Trust at the right level of security. While Arm and Microsoft are definitely not the only contributors in this field, we share the belief that a certain set of minimum requirements should be applied to IoT devices.

While there are similarities and differences, we believe our customers should be protected from the threats they are facing on their journey of digital transformation. We jointly call on the ecosystem – don't be disarmed by "not knowing where to start", instead use our guidelines to spearhead a more secure future.

References

1. PSA Certified, see psacertified.org
2. Microsoft Research; The Seven Properties of Highly Secured Devices (2nd Edition)
3. Ed Nightingale & Penny Orwick, Cybersecurity best practices to implement highly secured devices, see Cybersecurity best practices to implement highly secured devices – Microsoft Security Blog
4. Platform Security Model, see psacertified.org/app/uploads/2020/10/DEN0079_PSA_SM_BETA-0.pdf
5. PSA Functional API Certification, see psacertified.org/development-resources/certification-resources

Thank you.

psacertified™

Microsoft Azure