

Secure by Design Infographic Checklist

Is it secure? Your IoT security checklist

There's no escaping it – if you want to develop a robust, trustworthy product that will be secure during its total product lifecycle, then you need to ensure compliance with the UK's Secure by Design standards. The UK is taking 'decisive action' on IoT security and is taking a staged approach to enshrining the standards in law. The UK has also been leading efforts to create international alignment and as such has been working with other territories to align standards in the US (NIST) and EU (ETSI). In short, the direction of travel is only going one way.

If you're looking to design, develop or deploy a trustworthy, futureproof, secure IoT product, then our Secure by Design checklist can help.

Our checklist is by no means intended to be an exhaustive list, but instead a guide that gives you a set of questions to ask at every stage in the product development journey, perhaps throwing light on some important areas you may not have thought of yet. These questions are guided by four key aspects of attack surface analysis: deterrence, prevention, detection, correction and counter measures.

It's best used alongside our infographic: How to future-proof your IoT device – achieving 'Secure by Design' for IoT and connected devices.

// 13 Principles in the UK Government's Secure by Design Code of Practice

Secure by Design - Guidance at a glance

SbD1

No default passwords

All IoT device passwords must be unique and not resettable to any universal factory default value.

SbD2

Implement a vulnerability disclosure policy

Provide a public point of contact in order that security researchers and others are able to report issues.

SbD3

Keep software updated

All software components in internet-connected devices should be securely updatable.

SbD4

Securely store credentials and security-sensitive data

Any credentials must be stored securely within services and on devices. Hard-coded credentials in device software are not acceptable

SbD5

Communicate securely

Security-sensitive data, including any remote management and control, should be encrypted when transiting the internet.

SbD6

Minimise exposed attack surfaces

All devices and services should operate on the 'principle of least privilege'.

SbD7

Ensure software integrity

Software on IoT devices must be verified using secure boot mechanisms.

SbD8

Ensure personal data is protected

Where devices and/or services process personal data they should do so in accordance with data protection law.

SbD9

Make systems resilient to outages

Resilience must be built in to IoT services where required by the usage or other relying systems.

SbD10

Monitor system telemetry

If collected, all telemetry such as usage and measurement data should be monitored for security anomalies within it.

SbD11

Make it easy for users to delete personal data

Devices and services should be configured such that personal data can easily be removed or deleted by the consumer.

SbD12

Make installation and maintenance of devices easy

Installation and maintenance of IoT devices should employ minimal steps and should follow security best practice on usability.

SbD13

Validate Input data

Data input via user interfaces and transferred via application programming interfaces (APIs) or between networks must be validated.

→ **Access full definitions and details of the UK Government's Secure by Design principles here:**
https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/775559/

1

// The Beginning

Requirements Analysis

Your requirements analysis needs to capture all security requirements. But, there is invariably a dependency between the proposed user experience, and security to support that experience, which can be overlooked and needs requirement analysis. This leads to questions such as:

- What is the minimum viable solution that supports the user experience envisaged?
- How does departing from that minimum viable solution impact upon the requirements for security?
- What are the impacts of: installation, software updates, password entry, deletion of personal data and other security measures on the user experience?
- For each solution the security implementation should be proportionate to the Secure by Design guidance? Example - have the security requirements of the complete system been specified considering the extent to which data and communications need protecting? Has the impact on the user experience been considered for different security scenarios?
- Could the user experience be varied such that significant security requirements can be removed? Example – by not communicating or storing sensitive information the requirement to protect it can be removed.
- What are the impacts of negative security cases? Example – what is the user experience of failure to boot or if a breach is detected?
- Is the transfer of ownership a specified use case? What is the proposed approach to transferring ownership and resetting security?
- What happens when devices are decommissioned and disconnected from the service? Offboarding and onboarding are key moments in the user experience and have wide security implications.
- Has GDPR been adequately covered in the requirements analysis and security considerations?
- Are software updates forced, optional or deferrable? For different territories it may be necessary to consider who owns the product, interruption to usage and negative use cases of the update failing or taking an extended time due to communication issues?
- For the required functionality, how do the system components (silicon, radio, cloud, operating system, apps) communicate and what are their attack surfaces?
- What level of resilience is required? Example – what level of functionality is required if the product is disconnected?
- What telemetry is required to provide the required functionality and support security monitoring?
- What mandatory security standards apply to the solution (such as NIST or codes of practice) and have they been included in the requirements analysis?
- Has cloning of components been considered? What are the operational impacts of clone protection?
- Is key or certificate revocation required?
- Have the requirements for security patching all of the components in the system been considered? Depending upon the number and types of components there may be a complex set of patching updates that need to be managed.

2

// Before manufacture

Attack Surface Analysis

IoT systems often consist of multiple components with the ability to add and remove components. This introduces flexibility but also provides multiple attack points that can be exploited.

An attack surface analysis considers each of these components in isolation, considering their individual security requirements but also considering what happens as the system scales. Such an analysis leads to the questions below which can be addressed using threat analysis tools:

Threat Analysis Tools



Prevent: How can the implementation use standard security tool such as encryption or signing or one-time programming to prevent the threat?



Deter: What can be done to deter the threat?



Correct: How will the system respond and recover in order to correct the threat. Are countermeasures such as revocation or security renewal needed?



Detect: How will the threat be detected in an appropriate timescale?

STRIDE is a model of threats often used to identify and categorise computer security threats.

Threat	Desired property
Spoofing	Authenticity
Tampering	Integrity
Repudiation	Non-repudiability
Information disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Source: a model of threats developed by Praerit Garg and Loren Kohnfelder at Microsoft

- What physical and wireless network interfaces does my solution expose to the outside world?
- What hardware interfaces does my solution expose to the outside world?
- Have interfaces for servicing and refurbishing the product or any interface used during product development been considered? Examples – Jtag, serial and internal interfaces. Could these interfaces be secured?
- What are the protocols used for data in transit? Example – is HTTPS used?
- What storage is used for data at rest? Is this storage secure?
- What types of users and applications can talk to my solution? Is rights-based access implemented and if so what are the security requirements?
- How can I secure all of the interfaces for each user in a manner proportionate to my security requirements?
- What are the requirements for secure updates?
- What data am I expecting at each interface? What happens when that data is inconsistent with expectations?
- Removing complexity reduces risk, therefore, have I removed all libraries and drivers that are not required by my solution from the operating system?
- Does my solution require a secure trusted execution environment to isolate data and process code that is security critical?
- Is my system of such value that a Differential Power Analysis (DPA) attack is a concern for my product? Again, proportionate security requires consideration of the value of data and operation.
- Is complete disabling of the product required if tampering is detected? Example – the case is opened.

3

// Before manufacture

Design and Development

Interaction between the attack surface analysis and the design and development process uncovers realities around the proportionality of security requirements.

As the implications of the security requirements become apparent, decisions about what data is communicated, how it is stored and how the system operates will be clearer. This leads to questions such as:

- Has the design and development process introduced new interfaces, or protocols or other attack surfaces? If so, repeat the attack surface analysis from stage 2 for the emerging attack surfaces.
- If the design includes any keys or certificates, then what device security features will you need to provide an appropriate hardware root of trust?
- What operating system features does your design require? Example - should some processes have reduced privileges? Do you use memory protection, memory obfuscation or a trusted execution environment?
- What are the hardware implications of the secure boot requirements? Example – do you need a ROM boot load or an immutable flash participation?
- What are the hardware implications of the software update mechanism? Example - do you use double banked flash and what are the performance implications of writing to flash while maintaining normal operation?
- How do I use my attack surface analysis to harden my operating system and code?
- How many encryption keys do I need to secure various aspects of my design?
- What hardware and hardware design measures best suits my security needs? Example - consideration could include device selection based security features, package selection such as BGAs, buried tracks, test point coverage, blind vias, etc...
- How will I generate and distribute keys?
- Where is there any personal data stored in the system and is it secure?
- What happens when each of my interfaces disconnects – can my system deal with communication outages?
- Are my in-field devices uniquely identifiable and how is this identifier secured?
- What affordances need to be put in place to allow engineers to efficiently develop HW, SW and associated systems during development?
- Do development affordances create additional attack surfaces?
- Do the libraries and other software components used have any known, current vulnerabilities?
- What are the known vulnerabilities associated with selected protocols and encryption algorithms?
- Will the selected key lengths offer sufficient protection over the lifetime of the product?
- Is impaired network testing needed?
- Is penetration testing needed?
- Is fuzzy testing and negative use case testing required?

4

// Before manufacture

Production Planning

When considering system components, it's important to consider what is known about them. For most products the majority of software is third party code in the form of libraries and the OS. This is significant and can be an easily overlooked source of vulnerabilities. For components developed for the solution, the interaction between the attack surface analysis and the design and development process gives confidence that the system is well characterised. This is not the case for 'bought-in' components. Bought-in components must go through an attack surface analysis but this is likely to be dependent upon information from the supplier. So, the production/buying stage leads to questions such as:

- Am I building all of the solution or integrating some bought in components? If buying in components the attack surface analysis of stage 1 must be undertaken. This will require access to vulnerability disclosure from the component's supplier.
- If I am manufacturing, where am I manufacturing and how can I distribute keys to the site securely?
- How do I audit the manufacturing process to ensure security? How do I prevent cloning?
- Can keys be eavesdropped or manipulated during production and how will we detect this?
- How will keys, certificates and unique IDs be programmed into the product?
- When onboarding the device with cloud system how will keys the certificates be provisioned?
- What system integration and testing is required before I deploy?
- How will I check the correct programming of keys?
- Can production perform appropriate testing with security in place?
- How is production rework handled?

5

// During manufacture

Secure Key Generation

Depending upon the security requirements of your system there may be one or more secure keys required. This leads to questions such as:

- What elements of my system need symmetric keys?
- Are any of the keys shared between endpoints?
- Where am I generating the keys?
- Do I need and do I have a True Random Number Generator?
- What controls on people and equipment are needed during the key generation and distribution processes? Example – access to buildings, equipment and audit trails.
- How are the keys securely transferred to the endpoints?
- Are the keys stored securely at the endpoints including at the key generator if needed?
- Do keys expire and how are they updated?
- How will backup keys for disaster recovery and business continuity be managed?

6

// Operation & Updates

Consumer device in the field

Once deployed there are a myriad of requirements for managing and communicating with the device in the field. In field management is an essential aspect of maintaining your device security this leads to considerations such as:

- How do I record and track public key of unique ID's used in production?
- How will sensitive data be wiped when the product is decommissioned?
- Have the wider security implications of integrating the device into a system been considered?
- What other systems am I interoperating with and what are the security implications?
- Does the device management system support clone detection and have mechanisms to monitor telemetry for unusual behaviour?
- Are the operational and device management systems in place and appropriately secure for onboarding, status tracking, version control, ownership tracking, offboarding and device health?

7

// Operation & Updates

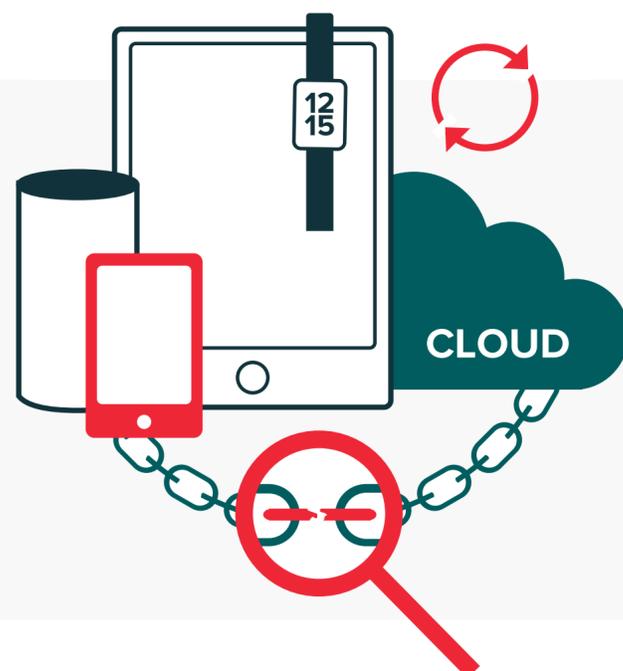
OTA Updates and Data Communication

Once deployed, the components in the field will begin communicating back and providing the system solution designed. At this stage communications between components and the ability to update them remotely becomes critical. Over the air updates (OTA) require management and occur with the secure environment implemented. This requires consideration of questions such as:

- What are my requirements for over the air updates and can my software be updated securely?
- What OTA software am I using and how secure is it?
- Can credentials and security sensitive data be stored securely?
- What feedback and success measurements are possible following update?
- How is failing hardware recorded?
- Does the security of the OTA and device management solution effectively lock you to that provider?

Secure by Design Infographic Checklist

Is it secure? Your IoT security checklist



//Total life cycle

Maintaining trust

Over time the solution must be kept healthy in the field; moving from fire-and-forget to management of the solution lifecycle. This ongoing responsibility has many facets and includes questions such as:

- Who monitors for emergent threats and provides fixes?
- Who monitors for security patches in my open source and third-party code and applies them?
- Do your suppliers have vulnerability disclosure policies that ensure you are notified of any vulnerabilities in their products and are they enforcing such a policy with their suppliers?
- Have countermeasure and remedial actions been considered for likely breaches?
- Who detects breaches and deploys countermeasures?
- Who writes and manages the vulnerability disclosure policy and do you have a procedure in place to respond to disclosures?
- Are security researchers able to easily and securely notify you about vulnerabilities?
- How will you communicate vulnerabilities to your customer base?
- How will you actively manage software updates?
- How will you repair secure devices in the field?
- What legal position should you take in regard to any breach?
- Do consumers of your products and services know how to contact you if they find a security vulnerability?
- Are you able to quantify the amount of cost/effort expected to process and resolve reported vulnerabilities in a timely manner?
- Do you have an internal procedure defined if a vulnerability is discovered which also may affect the wider industry?

Does the risk reward analysis mean that your business case stacks up?

Consult Red's experience in design and development of smart, connected devices and systems can reduce your time to market and manage cyber security risk to deliver truly secure IoT products.

Get in touch



Call us
+44 1274 287 700



Email us
info@consult.red



Read more
Security Page

© 2020 Red Embedded Consulting Ltd. Consult Red is a trading name of Red Embedded Consulting Ltd. Please refer to <https://consult.red/discover-red/> for more information. All rights reserved. This publication has been prepared for general guidance on matters of interest only and does not constitute professional advice. You should not act upon the information contained in this publication without obtaining specific professional advice. No representation or warranty (express or implied) is given as to the accuracy or completeness of the information contained in this publication, and, to the extent permitted by law, Red Embedded Consulting Ltd does not accept or assume any liability, responsibility or duty of care for any consequences of you or anyone else acting, or refraining to act, in reliance on the information contained in this publication or for any decision based on it.