Application Note: Mapping the IoT Security Foundation's Compliance Framework to the DCMS proposed Code of Practice for Security in Consumer IoT

March 7

2018

Applying technical controls from the IoT Security Compliance Framework to meet the DCMS proposed Code of Practice for Security in Consumer IoT Products and Associated Services

IoTSF Working Group Document

Applying Technical Controls Contained in the Compliance Framework To Meet the DCMS Code of Practice

The IoT Security Compliance Framework [Compliance Framework] [i] was first published in December 2016 by the IoT Security Foundation [IoTSF], and initially targeted at the Consumer/Smart Home markets. The Compliance Framework is a comprehensive checklist which guides a vendor through an assurance process, gathering evidence in a structured process and conforming to contemporary best practice and applicable standards.

On March 7th 2018, the UK's Department for Digital, Culture, Media and Sport (DCMS) published a report, Security by Design: Improving the cyber security of consumer Internet of Things (products and associated services) and a proposed "Code of Practice" [Code of Practice, CoP] [ii]

The aim of this application note is to show the mapping between the Compliance Framework and the Code of Practice.

The Code of Practice provides clear, top-level requirements that need to be met, yet translating these into practice can be technically complex. Many organisations developing IoT products are new to the world of product security design and management. They need a way to:

- Identify and understand the basic product / business security requirements
- Validate those security requirements have been considered / provisioned
- Ensure that security can be maintained over a life-cycle
- Communicate and verify all this to their customers.

This is the IoT Security Foundation's main objective: This application note is written for Device Manufacturers, IoT Service Providers, Mobile Application Developers and Retailers.

At the time of writing, IoTSF provides two primary sets of documents comprising:

- A set of clear and simple Best Practice Guides, for all departments in a company to use as an aide-memoire, defining what they need to do to build security into their products, services and operations. Example: Vulnerability Disclosure Guidelines [iii]

- The Compliance Framework, a checklist of all the items that management need to assure, both when a product is developed and put on the market and through its entire life-cycle, to make and keep it secure. These are drafted for every actor in the supply chain for IoT products and services, from the initial provider of technology components (such as processor cores and software modules) right through to the retailer and service provider. The Compliance Framework uses a common vocabulary to apply internally and to a supplier base, enabling a "supply chain of trust" to be communicated throughout the industry.

Cross Reference - Code of Practice and Compliance Framework 1.1

This section provides a detailed cross reference between the Code of Practice and version 1.1 of the Compliance Framework.

CoP No.	DCMS CoP Requirement	Req. No.	Compliance Framework Requirement
1	No default passwords: All IoT device passwords must be unique and not resettable to any universal factory default value.	2.4.7.7	If a connection requires a password or passcode or passkey for connection authentication, the factory issued or reset password is unique to each device and is not derived e.g. from serial numbers. Examples are WiFi access passwords and Bluetooth PINS.
		2.4.7.9	Where a wireless interface has an initial pairing process, the passkeys are changed from the factory issued or reset password prior to providing normal service.
		2.4.7.11	Where WPA2 WPS is used it has a unique, random key per device and enforces exponentially increasing retry attempt delays.
		2.4.8.3	Where a user interface password is used for login authentication, the factory issued or reset password is unique to each device in the product family.
		2.4.8.4	The product does not accept the use of null or blank passwords.
		2.4.8.5	The product will not allow new passwords containing the user account name with which the user account is associated.
		2.4.8.6	The product/system enforces passwords to be compliant as NIST SP800-63b [Section 5.1.1.2] or similar recommendations on: password length; characters from the groupings and special characters.
		2.4.8.12	The product allows the factory issued or OEM login accounts to be disabled, erased or renamed. This is to avoid the type of attacks where factory default logins and passwords are published on the web, which allows attackers to mount very simple scanning and dictionary attacks on devices.
		2.4.8.13	The product supports having any or all of the factory default user login passwords, altered prior to normal service. This is to avoid the type of attacks where factory default logins and passwords are published on the web, which allows attackers to mount very simple scanning and dictionary attacks on devices.
		2.4.10.4	Where a web user interface password is used for login authentication, the initial password or factory reset password is unique to each device in the product family.
		2.4.11.1	Where an application’s user interface password is used for login authentication, the initial password or factory reset password is unique to each device in the product family.

2	Implement a vulnerability disclosure policy	2.4.3.5	A policy has been established for dealing with both internal and third party security researcher(s) on the products or services.
		2.4.3.6	A security policy has been established for addressing changes, such as vulnerabilities, that could impact security and affect or involve technology or components incorporated into the product or service provided.
		2.4.3.7	Processes and plans are in place based upon the IoTSF "Vulnerability Disclosure Guidelines" or a similar recognised process to deal with the identification of a security vulnerability or compromise when they occur.
		2.4.3.8	A process is in place for consistent briefing of senior executives in the event of the identification of a vulnerability or a security breach, especially those who may deal with the media or make public announcements. In particular, that any public statements made in the event of a security breach should give as full and accurate an account of the facts as possible.
		2.4.3.9	There is a secure notification process based upon the IoTSF "Vulnerability Disclosure Guidelines" or a similar recognised process, for notifying partners/users of any security updates.
		2.4.3.11	As part of the Security Policy develop specific contact web pages for Vulnerability Disclosure reporting.
		2.4.3.12	As part of the Security Policy provide a dedicated security email address and/or secure webform for Vulnerability Disclosure communications.
		2.4.3.13	As part of the Security Policy develop a conflict resolution process for Vulnerability Disclosures.
		2.4.3.13	As part of the Security Policy publish the organisation's conflict resolution process for Vulnerability Disclosures.
		2.4.3.14	As part of the Security Policy develop response steps and performance targets for Vulnerability Disclosures.
		2.4.3.15	As part of the Security Policy develop security advisory notification steps.
		2.4.3.16	The Security Policy shall be compliant with ISO 30111 or similar standard.
3	Keep software updated	2.4.3.25	Where remote software upgrade can be supported by the device, there should be a published /transparent and auditable policy and schedule of actions to fix any vulnerabilities found.
		2.4.5.2	Where remote software upgrade can be supported by the device, the software images are digitally signed by the organisation's approved signing authority.
		2.4.5.3	A software update package has its digital signature, signing certificate and signing certificate chain verified by the device before the update

			process begins.
		2.4.5.4	If remote software upgrade is supported by a device, software images shall be encrypted whilst being transferred to it.
		2.4.5.8	The product has protection against reverting the software to an earlier and potentially less secure version.
		2.4.5.9	The cryptographic key chain used for signing production software is different from that used for any other test, development or other software images, to prevent the installation of non-production software onto production devices.
		2.4.5.10	Production software images should be assessed on release to remove all unnecessary debug and symbolic information “Know what is being released, and have checks in place to prevent accidental release of superfluous data
		2.4.5.11	Development software versions have any debug functionality switched off if the software is operated on the product outside of the product vendors’ trusted environment.
		2.4.6.2	Where remote update is supported, there is an established process/plan for validating and delivering updates on an on-going or remedial basis.
4	Securely store credentials and security sensitive data	2.4.5.7	The product’s software signing root of trust is stored in tamper-resistant memory.
		2.4.5.19	The production software signing keys are under access control.
		2.4.6.4	Files and directories are set to appropriate access privileges on a need to access basis.
		2.4.6.5	Passwords file(s) are owned by and are only accessible to and writable by the Devices’ OS’s most privileged account.
		2.4.6.8	The product’s OS kernel and its functions are prevented from being called by external product level interfaces and unauthorised applications.
		2.4.6.9	Applications are operated at the lowest privilege level possible.
		2.4.6.10	All the applicable security features supported by the OS are enabled.
		2.4.6.11	The OS is separated from the application(s) and is only accessible via defined secure interfaces.
		2.4.7.12	All network communications keys are stored securely, in accordance with industry standards such as FIPS 140 [5] or similar.
		2.4.8.1	The product contains a unique and tamper-resistant device identifier (e.g. such as the chip serial number or other unique silicon identifier) which is used for binding code and data to a specific device hardware.

		2.4.8.2	Where the product has a secure source of time there is a method of validating its integrity, such as Secure NTP. https://www.ntpsec.org/.
		2.4.8.8	The product securely stores any passwords using an industry standard cryptographic algorithm, compliant with an industry standard such as NIST SP800-63b [26] or similar.
		2.4.8.9	The product supports access control measures to the root account to restrict access to sensitive information or system processes.
		2.4.8.14	If the product has a password recovery or reset mechanism, an assessment has been made to confirm that this mechanism cannot readily be abused by an unauthorised party.
		2.4.8.16	The product allows an authorised factory reset of the device's authorisation information.
		2.4.9.4	There is a secure method of key insertion that protects keys against copying.
		2.4.9.7	The product stores all sensitive unencrypted parameters, (e.g. keys), in a secure, tamper-resistant location.
		2.4.9.9	In device manufacture all asymmetric encryption private keys that are unique to each device are secured in accordance with FIPS 140 [ref 5] and truly randomly internally generated or securely programmed into each device.
		2.4.11.5	The product securely stores any passwords using an industry standard cryptographic algorithm, for example see FIPS 140 [5].
		2.4.13.16	All the related servers and network elements store any passwords using a cryptographic implementation using industry standard cryptographic algorithms, for example see FIPS 140 [5].
		2.4.13.17	All the related servers and network elements support access control measures to restrict access to sensitive information or system processes to privileged accounts.
5	Communicate securely	2.4.5.15	The software must be architected to identify and ring fence sensitive software components, including cryptographic processes, to aid inspection, review and test. The access from other software components must be controlled and restricted to known and acceptable operations. For example security related processes should be executed at higher privilege levels in the application processor hardware.
		2.4.5.21	Where the device software communicates with a product related webserver or application over TCP/IP or UDP/IP, the device software uses certificate pinning or public/private key equivalent, where appropriate.
		2.4.5.22	The device remains secure and maintains state during a side channel

			attack.
		2.4.7.1	The product prevents unauthorised connections to it or other devices the product is connected to. For example is there a firewall on each interface and internet layer protocol.
		2.4.7.2	The network component and firewall (if applicable) configuration has been reviewed and documented for the required/defined secure behaviour
		2.4.7.4	Devices support only the latest versions of application layer protocols with no publically known vulnerabilities and it should not be possible to downgrade a connection to an older, less secure version.
		2.4.7.5	Insecure and unauthenticated application layer protocols (such as TELNET, FTP, HTTP, SMTP and NTP < v4) are not used.
		2.4.7.7	If a connection requires a password or passcode or passkey for connection authentication, the factory issued or reset password is unique to each device and is not derived e.g. from serial numbers.. Examples are WiFi access passwords and Bluetooth PINS.
		2.4.7.8	Where a wireless communications interface requires an initial pairing process, a Strong Authentication shall be used, requiring physical interaction with the device or possession of a shared secret. For example, Bluetooth Numeric Comparison.
		2.4.7.10	For any WiFi connection, WPA2 with AES or a similar strength encryption has been used and insecure protocols such as WPA and TKIP are disabled.
		2.4.7.11	Where WPA2 WPS is used it has a unique, random key per device and enforces exponentially increasing retry attempt delays.
		2.4.7.12	All network communications keys are stored securely, in accordance with industry standards such as FIPS 140 [5] or similar.
		2.4.7.13	Where the MQTT protocol is used, it is protected by a TLS connection with no known cipher vulnerabilities.
		2.4.7.14	Where the CoAP protocol is used, it is protected by a DTLS connection with no known cipher vulnerabilities.
		2.4.7.15	Where cryptographic suites are used such as TLS, all cipher suites shall be listed and validated against the current security recommendations such as NIST 800-131A 2] or OWASP. Where insecure ciphers suites are identified they shall be removed from the product.
		2.4.7.17	Where there is a loss of communications it shall not compromise the integrity of the device.
		2.4.7.18	The product only enables the communications interfaces, network protocols, application protocols and network services necessary for the products' operation.

		2.4.7.19	Communications protocols should be at the most secure versions available and/or appropriate for the product. For example, Bluetooth 4.2 rather than 4.0.
		2.4.7.20	Post product launch communications protocols should be maintained to the most secure versions available and/or appropriate for the product.
		2.4.9.1	A true random number generator source is exclusively used for all relevant cryptographic operations including nonce, initialisation vector and key generation algorithms. NIST SP 800-90A [3]
		2.4.9.2	The true random number generator source has been validated for true randomness using an NIST SP800-22 [4], FIPS 140-2 [5] or similar compliance process.
		2.4.9.3	There is a process for secure provisioning of keys that includes generation, distribution, revocation and destruction. For example in compliance with FIPS140-2 [5] or similar process.
		2.4.9.4	There is a secure method of key insertion that protects keys against copying.
		2.4.9.5	All the product related cryptographic functions have no publicly known unmitigated weaknesses, for example MD5 and SHA-1 are not used, e.g. those stipulated in NIST SP800-131A [2].
		2.4.9.6	All the product related cryptographic functions are sufficiently secure for the lifecycle of the product, e.g. those stipulated in NIST SP800-131A [2].].
		2.4.9.7	The product stores all sensitive unencrypted parameters, (e.g. keys), in a secure, tamper-resistant location.
		2.4.9.9	In device manufacture all asymmetric encryption private keys that are unique to each device are secured in accordance with FIPS 140 [5] and truly randomly internally generated or securely programmed into each device.
		2.4.11.4	Where the application communicates with a product related remote server(s) or device it does so over a secure connection such as a TLS connection using certificate pinning.
		2.4.12.2	The product/service ensures that all Personal Information is encrypted at rest and in transit.
		2.4.13.4	All the product related web servers' TLS certificate(s) are signed by trusted certificate authorities; are within their validity period; and processes are in place for their renewal.
		2.4.13.5	The Product Manufacturer or Service Provider has a process to monitor the relevant security advisories to ensure all the product related web servers use protocols with no publicly known weaknesses.

		2.4.13.6	The product related web servers support appropriately secure TLS/DTLS ciphers and disable / remove support for deprecated ciphers. For example those published at ENISA [27] SSL Labs [29], IETF RFC7525 [28]:
		2.4.13.7	The product related web servers have repeated renegotiation of TLS connections disabled.
		2.4.13.9	Where a product related to a webserver encrypts communications using TLS and requests a client certificate, the server(s) only establishes a connection if the client certificate and its chain of trust are valid.
		2.4.13.10	Where a product related to a webserver encrypts communications using TLS, certificate pinning is implemented. For example using OWASP, https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning or similar organisations' certificate and public key pinning guidance.
6	Minimise exposed attack surfaces	2.4.4.5	Any debug interface (for example, I/O ports such as JTAG) only communicate with authorised and authenticated entities on the production devices.
		2.4.4.9	All communications port(s), such as USB, RS232 etc., which are not used as part of the product's normal operation are not physically accessible or only communicate with authorised and authenticated entities.
		2.4.4.10	After manufacture, all the product's test points are securely disabled or removed wherever possible.
		2.4.5.1	The product has measures to prevent unauthenticated software and files being loaded onto it. In the event that the product is intended to allow un-authenticated software, such software should only be run with limited permissions and/or sandbox.
		2.4.5.5	If the product has any port(s) that are not required for normal operation, they are only allowed to communicate with authorised and authenticated entities or securely disabled when shipped. Where a port is used for field diagnostics, the port input is deactivated and the output provides no information which could compromise the device.
		2.4.5.10	Production software images should be assessed on release to remove all unnecessary debug and symbolic information "Know what is being released, and have checks in place to prevent accidental release of superfluous data
		2.4.5.11	Development software versions have any debug functionality switched off if the software is operated on the product outside of the product vendors' trusted environment.
		2.4.5.15	The software must be architected to identify and ring fence sensitive software components, including cryptographic processes, to aid inspection, review and test. The access from other software components must be controlled and restricted to known and

			acceptable operations. For example security related processes should be executed at higher privilege levels in the application processor hardware.
		2.4.5.19	The production software signing keys are under access control.
		2.4.6.3	All interactive OS accounts or logins have been disabled or eliminated from the software at the end of the software development process.
		2.4.6.4	Files and directories are set to appropriate access privileges on a need to access basis.
		2.4.6.5	Passwords file(s) are owned by and are only accessible to and writable by the Devices’ OS’s most privileged account.
		2.4.6.6	All OS non-essential services have been removed from the products’ software image or filesystems.
		2.4.6.7	All OS command line access to the most privileged accounts has been removed from the operating system.
		2.4.6.8	The product’s OS kernel and its functions are prevented from being called by external product level interfaces and unauthorised applications.
		2.4.6.9	Applications are operated at the lowest privilege level possible.
		2.4.6.11	The OS is separated from the application(s) and is only accessible via defined secure interfaces.
		2.4.7.1	The product prevents unauthorised connections to it or other devices the product is connected to. For example is there a firewall on each interface and internet layer protocol.
		2.4.7.2	The network component and firewall (if applicable) configuration has been reviewed and documented for the required/defined secure behaviour
		2.4.7.3	Products with one or more network interfaces, the uncontrolled, and any unintended packet forwarding function should be blocked.
		2.4.7.6	All the products unused ports are closed and the minimal required number of ports are active.
		2.4.7.18	The product only enables the communications interfaces, network protocols, application protocols and network services necessary for the products’ operation.
		2.4.7.19	Communications protocols should be at the most secure versions available and/or appropriate for the product. For example, Bluetooth 4.2 rather than 4.0.
		2.4.8.9	The product supports access control measures to the root account to restrict access to sensitive information or system processes.

		2.4.8.11	The product only allows controlled user account access; access using anonymous or guest user accounts are not supported without justification.
		2.4.13.2	Any product related web servers have their webserver identification options (e.g. Apache or Linux) switched off.
		2.4.13.3	All product related web servers have their webserver HTTP trace and trace methods disabled.
		2.4.13.6	The product related web servers support appropriately secure TLS/DTLS ciphers and disable / remove support for deprecated ciphers. For example those published at ENISA [27] SSL Labs [29], IETF RFC7525 [28]:
		2.4.13.7	The product related web servers have repeated renegotiation of TLS connections disabled.
		2.4.13.8	The related servers have unused IP ports disabled.
		2.4.13.17	All the related servers and network elements support access control measures to restrict access to sensitive information or system processes to privileged accounts.
		2.4.13.18	All the related and network elements servers prevent anonymous/guest access except for read only access to public information.
		2.4.14.1	The product has all of the production test and calibration software used during manufacture erased or removed or secured before the product is dispatched from the factory. This is to prevent alteration of the product post manufacture when using authorised production software, for example hacking of the RF characteristics for greater RF ERP. Where such functionality is required in a service centre, it shall be erased or removed upon completion of any servicing activities.
7	Ensure software integrity	2.4.4.1	The product's processor system has an irrevocable Secure Boot process.
		2.4.4.4	The Secure Boot process is enabled by default.
		2.4.8.2	Where the product has a secure source of time there is a method of validating its integrity, such as Secure NTP. https://www.ntpsec.org/.
		2.4.5.1	The product has measures to prevent unauthenticated software and files being loaded onto it. In the event that the product is intended to allow un-authenticated software, such software should only be run with limited permissions and/or sandbox.
		2.4.5.6	To prevent the stalling or disruption of the devices software operation any watchdog timers for this purpose cannot be disabled.
		2.4.5.7	The product's software signing root of trust is stored in tamper-resistant memory.

		2.4.5.8	The product has protection against reverting the software to an earlier and potentially less secure version.
		2.4.5.22	The device remains secure and maintains state during a side channel attack.
		2.4.5.24	The software has been designed to fail safely, i.e. in the case of unexpected invalid inputs, or erroneous software operation, the product does not become dangerous, or compromise security of other connected systems.
		2.4.14.5	Where a product includes a trusted secure boot process, the entire production test and any related calibration is executed with the processor system operating in its secured boot, authenticated software mode.
8	Ensure that personal data is protected	2.4.12.1	The product/service stores the minimum amount of Personal Information from users.
		2.4.12.2	The product/service ensures that all Personal Information is encrypted at rest and in transit.
		2.4.12.3	The product/service ensures that only authorised personnel have access to personal data of users.
		2.4.12.4	The product/service ensures that Personal Information is anonymised whenever possible and in particular in any reporting.
		2.4.12.5	The Product Manufacturer or Service Provider shall ensure that a data retention policy is in place, and compliant with the legal requirements for the territories the product or service is deployed.
		2.4.12.6	There is a method or methods for the product owner to be informed about what Personal Information is collected, why, where it will be stored.
		2.4.12.7	There is a method or methods for the product owner to check/verify what Personal Information is collected and deleted.
		2.4.12.8	The product / service can be made compliant with the local and/or regional Personal Information protection legislation where the product is to be sold.
		2.4.12.9	The supplier or manufacturer of any device shall provide information about how the device(s) functions within the end user’s network.
		2.4.12.10	The supplier or manufacturer of any devices or devices shall provide information about how the device(s) shall be setup to maintain the end user’s privacy and security.
		2.4.12.11	The supplier or manufacturer of any devices and/or services shall provide information about how the device(s) removal and/or disposal shall be carried out to maintain the end user’s privacy and security.

		2.4.12.12	The supplier or manufacturer of any devices or services shall provide clear information about the end user's responsibilities to maintain the devices and/or services privacy and security.
9	Make systems resilient to outages	2.4.13.20	Where a Product or Services includes any safety critical or life-impacting functionality, the services infrastructure shall incorporate protection against DDOS attacks, such as dropping of traffic or sink-holing. See NIST 800-53 SC-5 [32]
		2.4.13.21	Where a Product or Services includes any safety critical or life-impacting functionality, the services infrastructure shall incorporate redundancy to ensure service continuity and availability.
10	Monitor system telemetry data	N/A	This is covered in the IoTSF's Best Practice Guide K Logging.
11	Make it easy for consumers to delete personal data	2.4.12.5	The Product Manufacturer or Service Provider shall ensure that a data retention policy is in place, and compliant with the legal requirements for the territories the product or service is deployed.
		2.4.12.6	There is a method or methods for the product owner to be informed about what Personal Information is collected, why, where it will be stored.
		2.4.12.7	There is a method or methods for the product owner to check/verify what Personal Information is collected and deleted.
		2.4.12.8	The product / service can be made compliant with the local and/or regional Personal Information protection legislation where the product is to be sold.
		2.4.12.9	The supplier or manufacturer of any device shall provide information about how the device(s) functions within the end user's network.
		2.4.12.10	The supplier or manufacturer of any devices or devices shall provide information about how the device(s) shall be setup to maintain the end user's privacy and security.
		2.4.12.11	The supplier or manufacturer of any devices and/or services shall provide information about how the device(s) removal and/or disposal shall be carried out to maintain the end user's privacy and security.
		2.4.12.12	The supplier or manufacturer of any devices or services shall provide clear information about the end user's responsibilities to maintain the devices and/or services privacy and security.
		2.4.16.1	Where a device or devices are capable of having their ownership transferred to a different owner, all the previous owners Personal Information shall be removed from the device(s) and registered services. This option must be available when a transfer of ownership occurs or when an end user wishes to delete their Personal Information from the service or device.
		2.4.16.2	Where a device or devices user wishes to end the service, all that owners Personal Information shall be removed from the device and

			related services.
12	Make installation and maintenance of IoT devices easy	2.4.12.11	The supplier or manufacturer of any devices and/or services shall provide information about how the device(s) removal and/or disposal shall be carried out to maintain the end user's privacy and security.
		2.4.12.12	The supplier or manufacturer of any devices or services shall provide clear information about the end user's responsibilities to maintain the devices and/or services privacy and security.
		2.4.12.13	Security Usability: Devices and services should be designed with security usability in mind, reducing where possible, security friction and decision points that may have a detrimental impact on security. Best practices on usable security should be followed, particularly for user interaction and user interfaces.
13	Validate input data	2.4.10.1	Where the product or service provides a web based interface, Strong Authentication is used.
		2.4.10.10	All data being transferred over interfaces should be validated where appropriate. This could include checking the Data Type, Length, Format, Range, Authenticity, Origin and Frequency."
		2.4.10.11	Sanitise input in Web applications by using URL encoding or HTML encoding to wrap data and treat it as literal text rather than executable script
		2.4.10.12	All inputs and outputs are validated using for example a whitelist containing authorised origins of data and valid attributes of such data.
		2.4.11.7	All data being transferred over interfaces should be validated where appropriate. This could include checking the Data Type, Length, Format, Range, Authenticity, Origin and Frequency."
		2.4.11.9	All application inputs and outputs are validated using for example a whitelist containing authorised origins of data and valid attributes of such data see NIST SP 800-167 [34]

Table 1: Cross Reference of Code of Practice in Consumer IoT and the IoT Security Compliance Framework

References

i. IoT Security Compliance Framework Release 1.1, IoT Security Foundation. Latest release available here: https://www.iotsecurityfoundation.org/best-practice-guidelines/

ii. Security by Design: Improving the cyber security of consumer Internet of Things (products and associated services), DCMS, March 2018. www.gov.uk/government/publications/secure-by-design

iii. Vulnerability Disclosure Guidelines Release: 1.1, IoT Security Foundation. Latest release available here: https://www.iotsecurityfoundation.org/best-practice-guidelines/

[END]