Australian Government
Australian Signals Directorate

ACSC Australian Cyber Security Centre

# IoT Code of Practice – Guidance for Manufacturers

**First published:** September 2020
**Last updated:** October 2021

# Introduction

Adversaries regularly target Australian Government data in an attempt to gain an economic or strategic advantage. As such, Internet of Things (IoT) devices should have effective cyber security measures in place to defend against these threats.

The Australian Cyber Security Centre (ACSC) has produced this guidance for manufacturers to help them with the implementation of the thirteen principles outlined in the Australian Government's *Voluntary Code of Practice: Securing the Internet of Things for Consumers*. In doing so, this guidance focuses on devices, not their associated backend servers. Associated backend servers should instead follow their own better practice security guidance.

# The principles

## No duplicated default or weak passwords

### Examples of good implementation

- The device has a unique, unpredictable, complex and unfeasible to guess password for setup and access. Where this isn't possible, the device prompts users to set/change the password at first use.
- The default password is not publicly known or published.
- Where the user is prompted to set a password for the device or associated online account, the user is required to choose a password of at least the minimum length and complexity as articulated in the *Information Security Manual*.
- All online accounts associated with a device use WebAuthn or multi-factor authentication.
- The Wi-Fi access point hosted by the device, and used for setup, requires the user to authenticate.

### Examples of bad implementation

- The device has a weak default password that is unable to be changed.
- The device discloses its account password by simply interacting with it, without authenticating first.

## Implement a vulnerability disclosure policy

### Examples of good implementation

- A clear and detailed vulnerability disclosure policy is readily available on the manufacturer's website advising that lawyers won't be used to silence or prosecute people who report vulnerabilities in good faith.
- A point of contact specifically for reporting vulnerabilities is clearly identified.
- Vulnerabilities that are reported are acknowledged and responded to by the manufacturer.
- The manufacturer has an appropriately timely deadline for development and distribution of updates once vulnerabilities are identified.
- The manufacturer has a bug bounty program to encourage users to report vulnerabilities.
- The manufacturer has a service level agreement for users setting expectations and strategies to remediate or defend the device if a resolution for a vulnerability requires extensive engineering.
- Ownership of responsibility to defend the product is clearly identified.

## Keep software securely updated

### Examples of good implementation

- The manufacturer provides automatic updates and makes users aware via reliable communication (such as email or notification from an associated application) of when updates will be applied and what they will contain.
- The manufacturer digitally signs updates.
- The manufacturer always provides updates over secure network protocols.
- The manufacturer prioritises remediating design weaknesses and vulnerabilities, before introducing new features.
- Updates are easy for the user to find and apply.
- A change log is available to users detailing the purpose of each update.
- An end-of-life policy is available and visible to users.
- The device checks for updates both daily and on boot, and highlights the availability of updates to the user.
- The device validates updates through cryptographically-secure mechanisms prior to installation.

### Examples of bad implementation

- The device has vulnerabilities that the manufacturer claims to have patched that are still present following an update.
- Users are required to pay a subscription fee for updates, which is disproportionate to the type and cost of the device or service provided.
- The update makes changes to the device that reduce functionality and make users hesitant to apply updates.

## Securely store credentials

### Examples of good implementation

- Passwords stored on the device are encrypted using algorithms, as articulated in the *Information Security Manual*.
- All passwords and sensitive data exchanged during device setup are done so using cryptographically-secure mechanisms.

### Examples of bad implementation

- The device has hardcoded credentials that are contained within the firmware, and are common across multiple devices from the manufacturer.
- The device has obfuscated (e.g. base64 encoded) passwords that are stored in configuration or firmware rather than being encrypted using cryptographic algorithms.
- The device stores, shares or extracts passwords for Wi-Fi networks without suitable encryption or user interaction.
- Private key material is reused between multiple devices.

## Ensure that personal data is protected

### Examples of good implementation

- The manufacturer has a privacy policy that clearly describes personal data collected, including how it will be stored and used.
- Only personal data needed to operate the device is collected.
- Encryption, as articulated in the *Information Security Manual*, is used for data in transit and at rest.
- All personal data transmitted between the device and the mobile application is encrypted.
- Metadata and content associated with sensitive personal data is encrypted.
- The manufacturer complies with relevant regulatory and statutory requirements (e.g. *Australian Privacy Principles* and *Australian Consumer Law*).

### Examples of bad implementation

- Sensitive personal data is transmitted in clear text.
- The privacy policy contradicts elements of the *Australian Privacy Principles*.
- The user is encouraged to connect with third party online platforms for the purpose of sharing data without being informed about the implications of the sharing arrangement.

## Minimise exposed attack surfaces

### Examples of good implementation

- During setup, the device only opens required physical interfaces or network ports.

- After the setup process is complete, physical interfaces or network ports that were only required for device setup are closed.
- Physical interfaces or network ports required for configured functionality are only exposed when the user has securely configured that functionality.
- Manufacturing and debug interfaces (e.g. JTAG and UART) are disabled on production hardware.
- Software, including plugins and extensions, operates under the principle of least privilege.
- The web management interface is only accessible to the local network unless the device needs to be managed remotely via the internet.
- A secure software development process is implemented and includes penetration testing.
- Backups of configuration data are only available after authentication.

### Examples of bad implementation

- The device has multiple unused network ports that are open and listening for connections both before and after device setup.
- Bluetooth pairing stays active once the device has been set up despite no longer being used for any device functionality.
- The device exposes unencrypted protocols (e.g. Telnet) which are used to exchange usernames and passwords that gain root access to the device.
- The device has unused physical interfaces (e.g. USB ports) available.

## Ensure communication security

### Examples of good implementation

- Data, especially sensitive data, in transit to and from the device is encrypted, as articulated in the *Information Security Manual*.
- Encryption is used for communication during device setup.
- Logs detailing remote access to the device are provided to the user.
- All remote access to the device is encrypted.

### Examples of bad implementation

- Encoding, rather than encryption, is used to protect sensitive data, including: usernames, passwords, authentication tokens and other forms of credentials.
- The Wi-Fi SSID and password are sent in plaintext between the application and the device during setup.
- The user's activity in the application is transferred unencrypted across a network.

## Ensure software integrity

### Examples of good implementation

- The device requires updates to be digitally signed by the manufacturer.
- The device has a boot mechanism that checks for errors and notifies the user of any failed updates.

- The manufacturer includes plugins or extensions to the device in security reviews.
- The device only allows use of signed plugins or extensions.
- Plugins or extensions only communicate with online resources using secure mechanisms which confirm the integrity of remote resources.
- The manufacturer provides change logs to identify changes to the device.

## Make systems resilient to outages

### Examples of good implementation

- The device powers on and reconnects to the network without any user interaction following unexpected loss of power.
- The user is notified immediately about the device going offline when network connectivity is lost.
- The device reconnects to the network automatically and notifies the user once the device reconnects.
- Battery backups are provided for relevant devices, which activate on loss of power.
- Secure alternative communication mechanisms (e.g. Bluetooth and Wi-Fi) are provided in the event of network loss.
- The device maintains essential functionality if network connectivity is unavailable.

## Monitor system telemetry data

### Examples of good implementation

- The user is informed whether or not telemetry data is collected.
- Telemetry data collected is only used to improve device functionality, integrity, security and monitor for security anomalies.

## Make it easy for consumers to delete personal data

### Examples of good implementation

- The user can easily delete their personal data, at no cost, from both their device and the associated online accounts.
- Processes for deleting personal data are clearly explained on the manufacturer's website.
- The device has a user-friendly factory reset process that is clearly outlined in the device manual, is easy to complete and includes the removal of user-provided data and configuration.
- The manufacturer allows the user to delete single pieces of data while maintaining other data.
- All personal data is securely deleted upon removal of an account.
- All configuration data is securely deleted at the request of the user.

### Examples of bad implementation

- The only way to delete personal data is through 'ageing' (i.e. waiting for the advertised storage period to elapse).

- The user's data is maintained on their device even after the user intentionally attempts to delete it, and believes the data has been deleted.
- The user is required to contact the manufacturer in order to delete personal data.

## Make installation and maintenance of devices easy

### Examples of good implementation

- The manufacturer provides tips for the user on how to setup the device securely.
- The manufacturer explains how to use the device through a short video or interactive demo.
- Installation and secure configuration of the device is easy.
- Device documentation clearly describes how to install and configure the device.
- The application or online interface provides a step-by-step setup process.
- Device documentation clearly matches the actual user experience with the device.
- Updates are applied automatically to maintain security throughout the device's life.
- Where continuous battery operation is expected, the device immediately notifies the user when the batteries are running low.

### Examples of bad implementation

- Installation causes issues that require a factory reset to remedy them.
- The device restarts during setup in order to connect, and fails to notify the user if any issues have occurred during this process.
- Maintaining the device requires substantial user investment in training or education that does not suit the intended users or usage of the device.

## Validate input data

### Examples of good implementation

- Input of data by a user requires authentication.
- Invalid and non-authorised input data are rejected.
- The manufacturer follows best practice advice in order to reduce the attack surface of strings designed to encode or carry content beyond expected user input (e.g. rejects special characters, escape characters, non-ASCII or Unicode).
- The device defends itself against exploitation techniques such as SQL injection.
- Both client-side validation and server-side validation are performed.

### Example of bad implementation

- The device crashes due to unexpected input.

# Further information

The *Information Security Manual* is a cyber security framework that organisations can apply to protect their systems and data from cyber threats. The advice in the *Strategies to Mitigate Cyber Security Incidents*, along with its Essential Eight, complements this framework.

# Contact details

If you have any questions regarding this guidance you can write to us or call us on 1300 CYBER1 (1300 292 371).