Software Security Code of Practice - Assurance Principles and Claims (APCs)

Helps vendors measure how well they meet the Software Security Code of Practice, and suggests remedial actions should they fall short.

About this document

This document allows technical staff to self-assess their own software practices against the Software Security Code of Practice, a set of outcomes that – if met – means the vendor can claim that their software has achieved a baseline of software security and resilience.

This document considers each principle within the four themes in the Code, and breaks them down into a set of **individual claims** that describe a way to fully meet the associated principle. If all these claims are well-evidenced, then a vendor can claim in good faith that they are meeting the principle.

Note that:

- The type of evidence can vary but examples include document inspection, interviewing individuals, or auditing test plans and results.
- Vendors requiring independent audit of compliance with the code should contact any NCSC-approved cyber resilience test facility.
- Where users recognise that they are unable to evidence a claim fully then it should be clear what steps to take to correct this. For example, certain processes or information may need to be more fully documented, or additional testing may need to be put in place.
- If it is unclear whether a form of evidence is suitable, vendors can refer to the claims trees in the appendix. The claim trees use a claims argument evidence method, similar to that used in the NCSC Principles Based Assurance (PBA) method.

Theme 1: Secure design and development

This theme ensures that software is secure when first provided to customers, along with subsequent updates. The use of appropriate design and development practices reduces the likelihood of errors and the presence of vulnerabilities in software and updates.

Principle 1.1	Claims
Follow an established secure development framework.	• The development framework used is documented. • Developers are trained in the use of the framework and tools. • Tools are maintained and updated. • Items requiring configuration control are identified and version control is used. • Requirements are captured and recorded. • Software is designed for user need.

Principle 1.2	Claims
Understand the composition of the software and assess risks linked to the ingestion and maintenance of third-party components throughout the development lifecycle.	• All third-party components are identified and documented. • Integrity of third-party components and updates is verified. • Each third-party component is tested before being first deployed. • Third-party component updates are tested. • Processes are in place to manage and deploy updates to third-party components.

Principle 1.3	Claims
Have a clear process for testing software and software updates before distribution.	• A test plan exists that covers all requirements and third-party components. • Execution of the test plan is automated and repeatable wherever possible. • Defects identified during testing are addressed.

Principle 1.4	Claims

Follow secure by design and secure by default principles throughout the development lifecycle of the software.	• Techniques to understand how the software might be exploited (threat modelling) have been used in the design of the software. • Multi-factor authentication for privileged users of the software is enforced. • Default (and persistent) passwords are not used. • Data input to the software is validated. • Credentials and sensitive data are securely stored.

Theme 2: Build environment security

This theme ensures appropriate steps are taken to prevent the build environment from being accessed by a person (or machine) without a legitimate need. This helps prevent interference with the software during the build and release process. The ‘build environment’ refers to the environment where software is *compiled*, *built* and *packaged* ready for release. This should be logically or physically separate from areas where code is *written* and *tested*.

Principle 2.1	Claims
Protect the build environment against unauthorised access.	• Roles are defined that specify the data and functionality that each role is allowed to access. • Users of the build environment are required to authenticate on a regular basis. • Users of the build environment are issued with credentials bound to their role. • Credentials are securely managed and stored. • Credentials are multi factor. • Users with access to the build environment are regularly reviewed to ensure they still have a legitimate need.

Principle 2.2	Claims
Control and log changes to the build environment.	• Access and changes to the build environment are logged. • Only authorised personnel can make changes to the build environment. • Logs are auditable and retained for an agreed period. • The confidentiality and integrity of logs is protected.

Theme 3: Secure deployment and maintenance

This theme ensures that the software remains secure throughout its lifetime. The use of appropriate mechanisms and processes for managing and deploying updates to software reduces the likelihood of errors and vulnerabilities persisting in software.

Principle 3.1	Claims
Distribute software securely to customers.	• The integrity of software (including updates) can be verified in the customer environment. • Software (including updates) is distributed over trusted channels.

Principle 3.2	Claims
Implement and publish an effective vulnerability disclosure process.	• A vulnerability disclosure policy and process is published. • The vulnerability disclosure process describes how to confidentially report vulnerabilities.

Principle 3.3	Claims
Have processes and documentation in place for proactively detecting, prioritising and managing vulnerabilities in software components.	• Knowledge of public vulnerabilities is kept up to date. • A vulnerability management plan exists that assesses and prioritises responses to vulnerabilities.

Principle 3.4	Claims
Report vulnerabilities to relevant parties where appropriate.	• Internal security teams are informed. • Affected customers are informed.

Principle 3.5	Claims

Provide timely security updates, patches and notifications to customers.	• Security updates are distributed as soon as is practicable. • Security updates are tested and secure by default.

Themes 4: Communication with customers

This theme ensures that vendors provide sufficient information to customers to enable them to effectively manage risks and incidents throughout the software's lifetime.

Principle 4.1	Claims
Provide information to the customer specifying the level of support and maintenance provided for the software being sold.	• 'End of support' dates are published for all software components. • A policy on frequency of updates and the process for applying them is published. • User documentation describes how to correctly and securely apply updates and use software.

Principle 4.2	Claims
Provides at least 1 year's notice to customers of when the software will no longer be supported or maintained by the vendor.	• Customers are given at least 1 year's notice of when software will no longer be supported.

Principle 4.3	Claims
Make information available to customers about notable incidents that may cause significant impact to customer organisations.	• An incident support plan is published. • Customers are informed of relevant incidents in a timely manner.

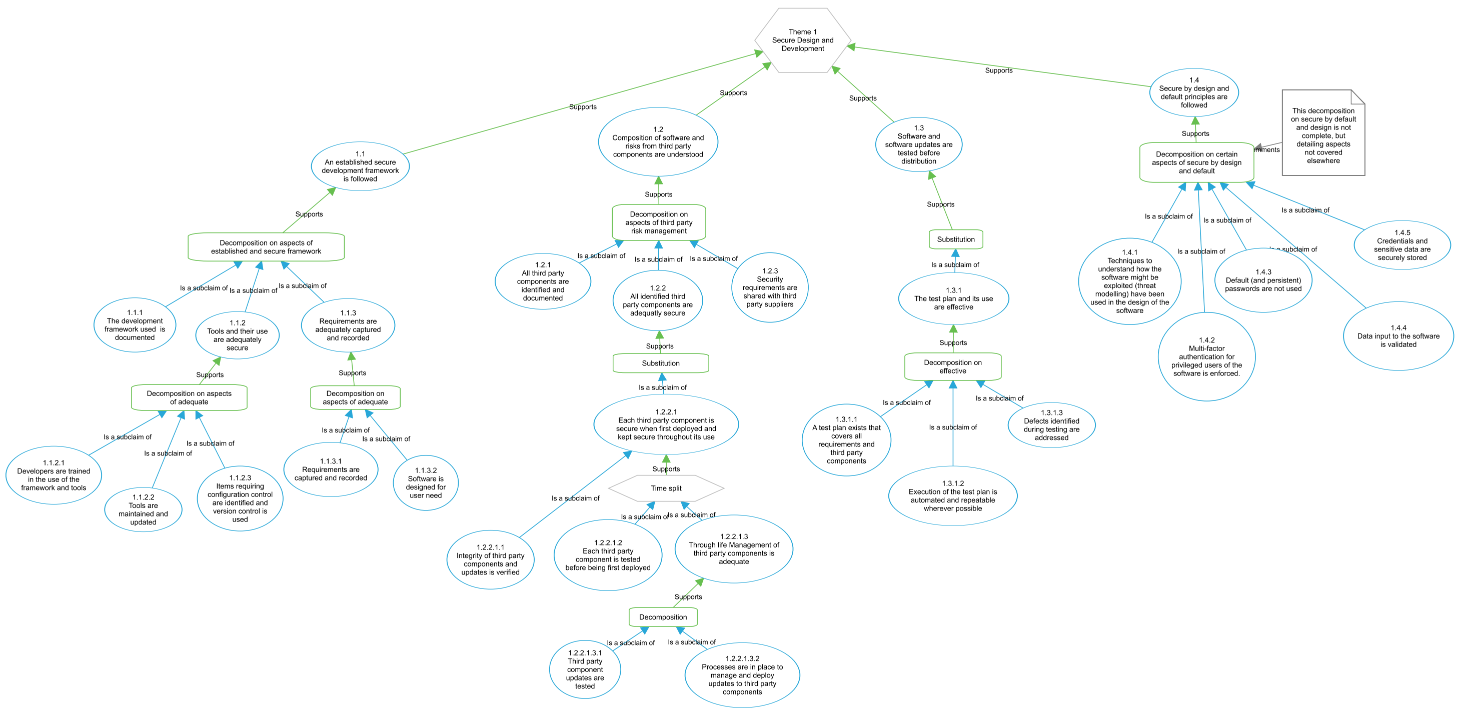
Appendix: Claims trees

The following claims trees display the reasoning behind the claims in the body of this document. The **themes** are presented as stated in the code whilst the **principles** are expressed as 'top-level claims' that enable the use of the claims argument evidence method to further deconstruct top-level claims to 'low-level claims' that can be simply and objectively evidenced.

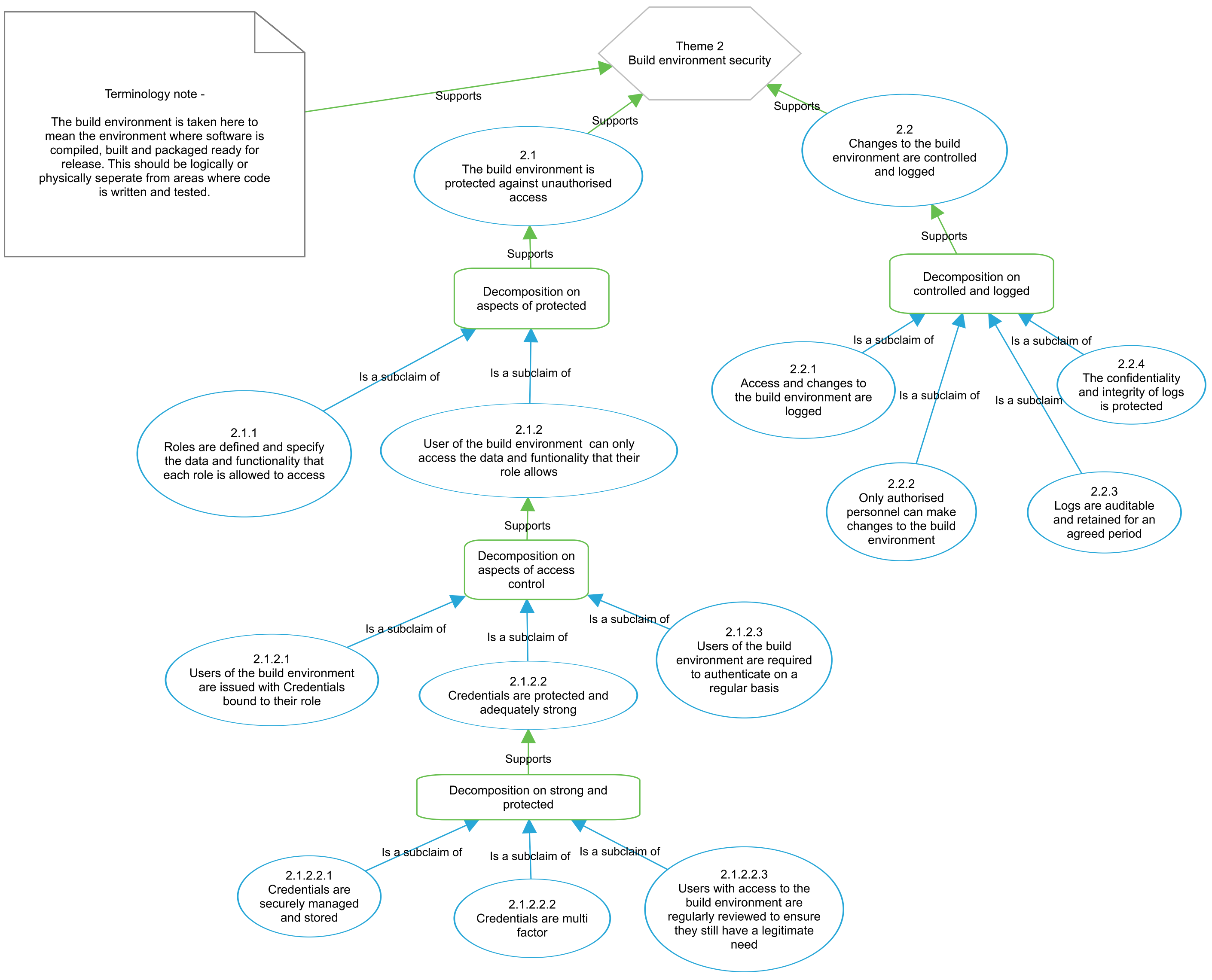
If a vendor is unclear on the intention of one of the claims in this document, then the claims tree can be useful in gaining an understanding of the provenance and rationale for the claim in relation to the principles.

Similarly, if a user cannot fully evidence a claim, then the claims tree can help in understanding the implications of this (for example as risk against the underlying principle), and the changes or improvements that can be made to help develop evidence in the future.

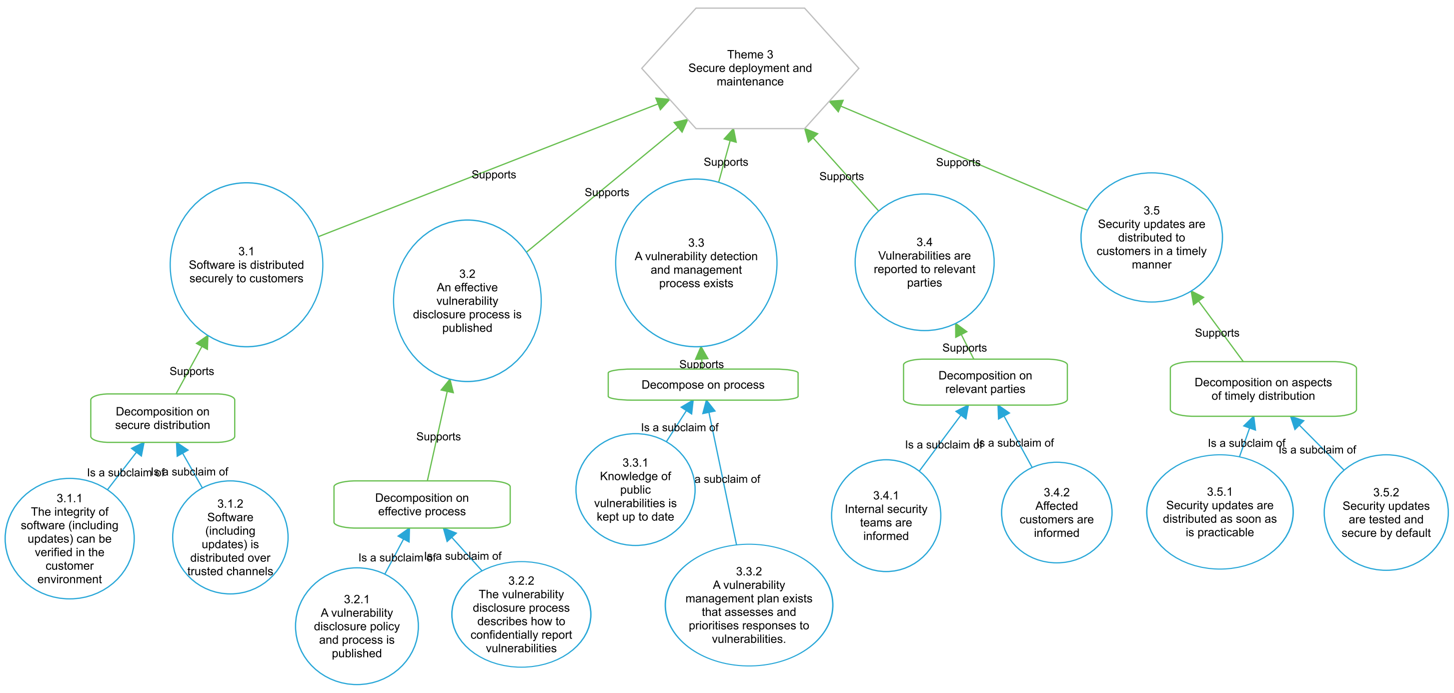
Theme 1: Secure design and deployment


Theme 1 Secure Design and Development
Supports
Is a subclaim of
1.1 An established secure development framework is followed
Decomposition on aspects of established and secure framework
1.1.1 The development framework used is documented
1.1.2 Tools and their use are adequately secure
1.1.3 Requirements are adequately captured and recorded
Decomposition on aspects of adequate
1.1.2.1 Developers are trained in the use of the framework and tools
1.1.2.2 Tools are maintained and updated
1.1.2.3 Items requiring configuration control are identified and version control is used
Decomposition on aspects of adequate
1.1.3.1 Requirements are captured and recorded
1.1.3.2 Software is designed for user need
1.2 Composition of software and risks from third party components are understood
Decomposition on aspects of third party risk management
1.2.1 All third party components are identified and documented
1.2.2 All identified third party components are adequatly secure
1.2.3 Security requirements are shared with third party suppliers
Substitution
1.2.2.1 Each third party component is secure when first deployed and kept secure throughout its use
Time split
1.2.2.1.1 Integrity of third party components and updates is verified
1.2.2.1.2 Each third party component is tested before being first deployed
1.2.2.1.3 Through life Management of third party components is adequate
Decomposition
1.2.2.1.3.1 Third party component updates are tested
1.2.2.1.3.2 Processes are in place to manage and deploy updates to third party components
1.3 Software and software updates are tested before distribution
Substitution
1.3.1 The test plan and its use are effective
Decomposition on effective
1.3.1.1 A test plan exists that covers all requirements and third party components
1.3.1.2 Execution of the test plan is automated and repeatable wherever possible
1.3.1.3 Defects identified during testing are addressed
1.4 Secure by design and default principles are followed
Decomposition on certain aspects of secure by design and default
This decomposition on secure by default and design is not complete, but detailing aspects not covered elsewhere
1.4.1 Techniques to understand how the software might be exploited (threat modelling) have been used in the design of the software
1.4.2 Multi-factor authentication for privileged users of the software is enforced.
1.4.3 Default (and persistent) passwords are not used
1.4.4 Data input to the software is validated
1.4.5 Credentials and sensitive data are securely stored

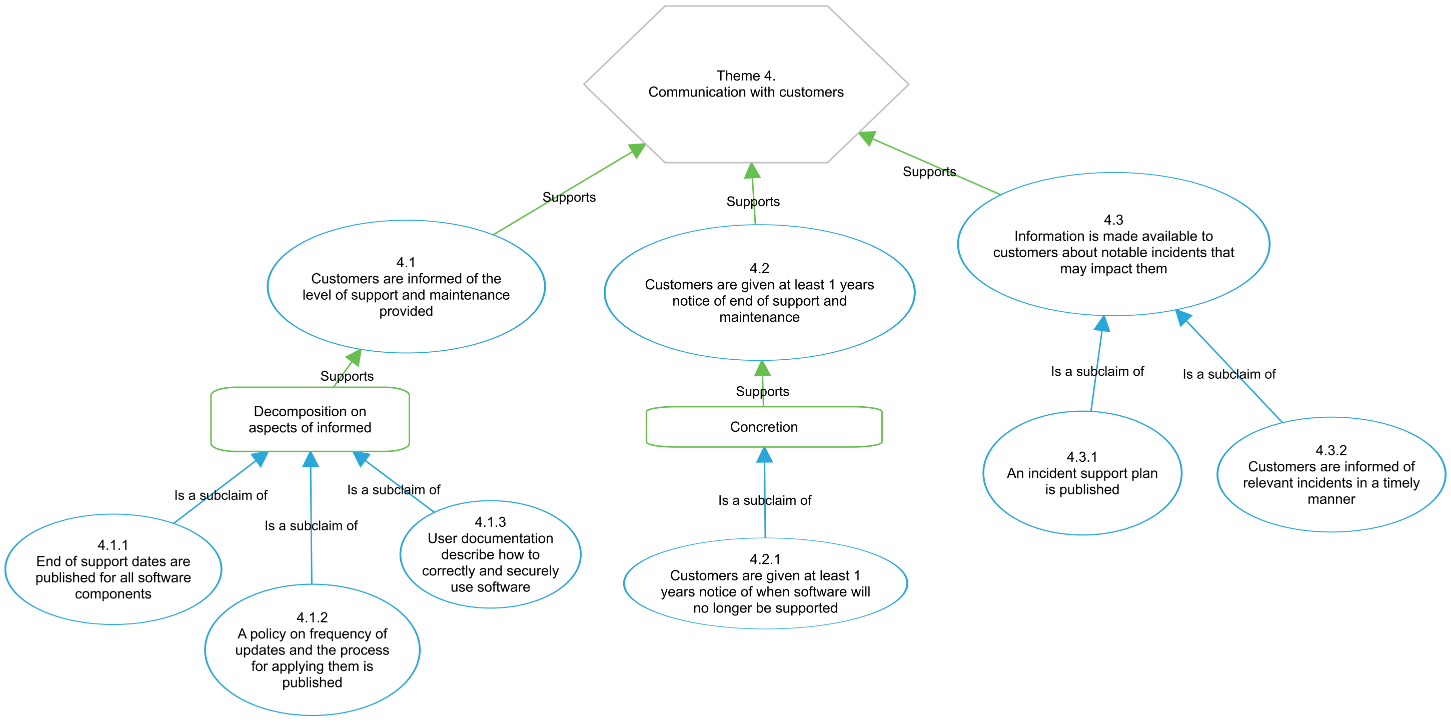
Theme 2: Build environment security


Terminology note -
The build environment is taken here to mean the environment where software is compiled, built and packaged ready for release. This should be logically or physically seperate from areas where code is written and tested.
Supports
Theme 2
Build environment security
Supports
Supports
2.1
The build environment is protected against unauthorised access
2.2
Changes to the build environment are controlled and logged
Supports
Decomposition on aspects of protected
Supports
Decomposition on controlled and logged
Is a subclaim of
Is a subclaim of
2.1.1
Roles are defined and specify the data and functionality that each role is allowed to access
2.1.2
User of the build environment can only access the data and funtionality that their role allows
Is a subclaim of
Is a subclaim of
Is a subclaim of
Is a subclaim
2.2.1
Access and changes to the build environment are logged
2.2.4
The confidentiality and integrity of logs is protected
2.2.2
Only authorised personnel can make changes to the build environment
2.2.3
Logs are auditable and retained for an agreed period
Supports
Decomposition on aspects of access control
Is a subclaim of
Is a subclaim of
Is a subclaim of
2.1.2.1
Users of the build environment are issued with Credentials bound to their role
2.1.2.2
Credentials are protected and adequately strong
2.1.2.3
Users of the build environment are required to authenticate on a regular basis
Supports
Decomposition on strong and protected
Is a subclaim of
Is a subclaim of
Is a subclaim of
2.1.2.2.1
Credentials are securely managed and stored
2.1.2.2.2
Credentials are multi factor
2.1.2.2.3
Users with access to the build environment are regularly reviewed to ensure they still have a legitimate need

Theme 3: Secure deployment and maintenance


Theme 3
Secure deployment and maintenance
Supports
3.1
Software is distributed securely to customers
Supports
Decomposition on secure distribution
Is a subclaim of
Is a subclaim of
3.1.1
The integrity of software (including updates) can be verified in the customer environment
3.1.2
Software (including updates) is distributed over trusted channels
3.2
An effective vulnerability disclosure process is published
Supports
Decomposition on effective process
Is a subclaim of
Is a subclaim of
3.2.1
A vulnerability disclosure policy and process is published
3.2.2
The vulnerability disclosure process describes how to confidentially report vulnerabilities
3.3
A vulnerability detection and management process exists
Supports
Decompose on process
Is a subclaim of
Is a subclaim of
3.3.1
Knowledge of public vulnerabilities is kept up to date
3.3.2
A vulnerability management plan exists that assesses and prioritises responses to vulnerabilities.
3.4
Vulnerabilities are reported to relevant parties
Supports
Decomposition on relevant parties
Is a subclaim of
Is a subclaim of
3.4.1
Internal security teams are informed
3.4.2
Affected customers are informed
3.5
Security updates are distributed to customers in a timely manner
Supports
Decomposition on aspects of timely distribution
Is a subclaim of
Is a subclaim of
3.5.1
Security updates are distributed as soon as is practicable
3.5.2
Security updates are tested and are secure by default

Theme 4: Communication with customers


Theme 4.
Communication with customers
Supports
Supports
Supports
4.1
Customers are informed of the level of support and maintenance provided
Supports
Decomposition on aspects of informed
Is a subclaim of
Is a subclaim of
Is a subclaim of
4.1.1
End of support dates are published for all software components
4.1.2
A policy on frequency of updates and the process for applying them is published
4.1.3
User documentation describe how to correctly and securely use software
4.2
Customers are given at least 1 years notice of end of support and maintenance
Supports
Concretion
Is a subclaim of
4.2.1
Customers are given at least 1 years notice of when software will no longer be supported
4.3
Information is made available to customers about notable incidents that may impact them
Is a subclaim of
Is a subclaim of
4.3.1
An incident support plan is published
4.3.2
Customers are informed of relevant incidents in a timely manner

PUBLISHED

7 May 2025

REVIEWED

7 May 2025

VERSION

1.0

WRITTEN FOR

Small & medium sized organisations

Large organisations

Public sector

Cyber security professionals