National Cyber Security Centre

a part of GCHQ

Product Cyber Resilience Testing (CRT) Assurance Principles and Claims

Version 1.0

April 2025

Document History

Date	Version	Change history details	SharePoint Ref
24/04/2025	1.0	Final Version	NCSC-1015903574-1672

A review will take place October – December of each year for publication the following February.

Document owner

1. Introduction

Product Cyber Resilience Testing

The NCSC Product Cyber Resilience Testing service (CRT) is designed to assess a connected products resilience to attack from its public facing connections (most notably connections to the internet). As such CRT can usefully be applied to any connected product to provide confidence that it can protect itself when faced with malicious activity from public domains.

CRT does not seek to assess how well a product performs it's intended function (security or otherwise), these should be assessed using other assurance services designed for these purposes.

Assurance Principles and Claims (APC) documents

An APC document is the key artefact for any assurance service using the NCSC Principles Based Assurance (PBA) method. The APC plays the role of a standard in traditional compliance-based assurance activity, defining the scope of the assurance activity and the information needed for an assessment.

The APC is a collection of Principles and Claims. The principles define the scope of the assurance activity setting out an ideal state that a product can be measured against. To assist with this measurement each principle is deconstructed into a set of claims (using a claims-argument-evidence method) that are designed to be easily evidenced.

Concepts of CRT

CRT is built on the idea that to be resilient to a public connection a product should be designed to be secure, be well engineered and maintained, and also be able to protect itself from public connections in usage and operation.

Good engineering and maintenance of a product helps build confidence that exploitable vulnerabilities and other weaknesses in the product are minimised during initial development and managed down throughout a product's operational life. It is also essential that a product is designed to promote secure use and to manage its public connections in a way that prevents malicious actors gaining access to (or interfering with) it without the need to exploit vulnerabilities.

Assumptions and structure of CRT

Although CRT can be applied to any connected product the approach encapsulated in this APC assumes that such products will have significant software elements, and that that when faced with a commodity threat model it is this software, together with its use and operation, that present the most likely attack surfaces. Accordingly, this APC focuses on these 2 aspects and builds on the APC published in support of the Software security Code of Practice.

When faced with an elevated threat model it is also appropriate to consider the attack surface that any developed hardware elements may present; developers of such products should consult NCSC or an approved CRTF if they want such an assessment.

The principles are grouped under 5 themes. The first 4 themes are, by design, identical to the themes in the Software security Code of Practice, and if developers have been previously assessed against these themes by a CRTF they do not need to resubmit evidence as part of a product assessment. They should however state that previous evidence is relevant to the product being assessed. A previous assessment against themes 1 to 4 may have been conducted as a standalone assessment against the Code of Practice APC or as part of another product assessment.

Themes 1 to 4 principles and claims are reproduced below from the Code of Practice APC purely to allow this document to operate as a standalone APC.

Theme 5 is unique to this APC and designed to allow the gathering of evidence against product specific features. In some case these explore areas not covered by the Code of Practice (eg usability), and in other areas they explore further some of the aspects touched on in themes 1 to 3, e.g. around secure by design and operation features. As such re-use of evidence between themes is acceptable.

Assessing a product using this APC

The APC considers each principle in turn 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. The claims are derived using a claims argument evidence method. The reasoning used for each claim is presented as a graphical claims tree in the appendix.

Note that the type of evidence can vary but examples include document inspection, interviewing individuals, or auditing test plans and results. Self-assessment is possible, but vendors requiring independent audit or validation of evidence 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.

Where a claim is not judged to be fully met the assessment should interpret the impact of this against the underlying principle (usually as a risk). The claims trees in the appendix will be useful in developing this interpretation as they provide a reasoned link from each claim back to the principle.

2. Themes, Principles and Claims

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.

Theme 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.

Theme 5: Product specific usage, design and operation

This theme focuses in more detail on the specific features of the product being assessed. A product that does not adequately support all users in its secure operation can be explored without the need for sophisticated attacks. This theme also explores further security critical aspects of secure by design and operation, particularly around authentication when accessing the product, and the product ability to protect its own data and integrity, and recognise and respond to security incidents.

Principle 5.1	**Claims**
Design the product to be usable	• The product's usability has been demonstrated to support people in its secure installation, use and maintenance • Guidance and support on product configuration is available to those installing and using the product • People have a mechanism to report difficulties in working with the product to the developers • If the product is reset it defaults to a known safe state • Accessibility testing demonstrates the design supports disabled people in securely installing, using and maintaining the product • People are notified if the product is insecurely configured • The product is designed to allow people to easily recover from errors
Principle 5.2	**Claims**

Ensure only authorised users have access to product data and functionality	• Users are only allowed access to the data and functionality necessary to perform their role • Logging and auditing of access to the product is in place • Users are required to enter their credentials before accessing any data or functionality • Credentials are managed securely • Privileged roles are defined and credentials for such roles are multi-factor • Credentials are unique per device or defined by the user when first used • There is a way for users to recover from loss of credential
Principle 5.3	**Claims**
Protect sensitive data and the integrity of the product	• All types of sensitive data are identified and documented • Sensitive data is only transmitted to/from trusted connections • The confidentiality and integrity of sensitive data is protected during transmission • Where the product uses encryption a recognised cryptographic standard is used • The integrity of software or firmware is verified when loaded • Remote management commands that affect product operation are verified before being acted upon
Principle 5.4	**Claims**
Enable the logging and monitoring of security events	• All security events are defined • When a security event occurs it is logged • The format of logging data is defined • The integrity of logging data is protected • Logs can only be accessed by authorised users

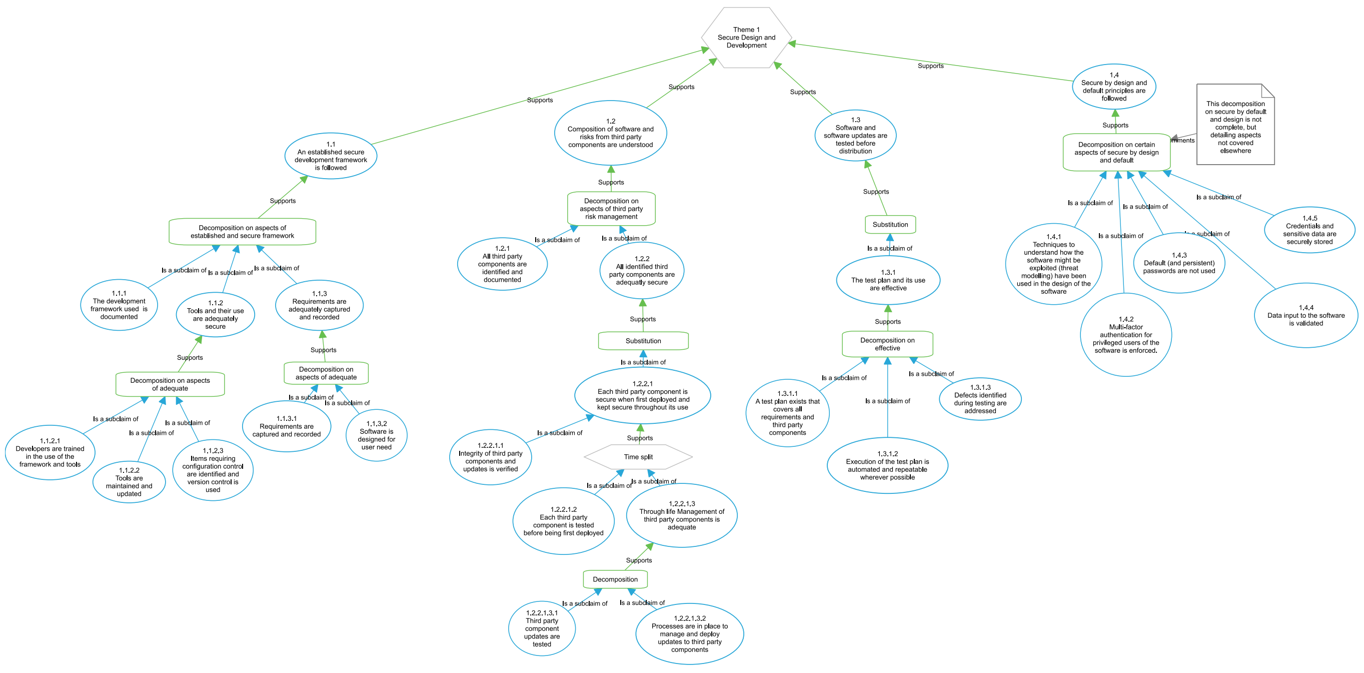
Appendix: Claims trees

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.

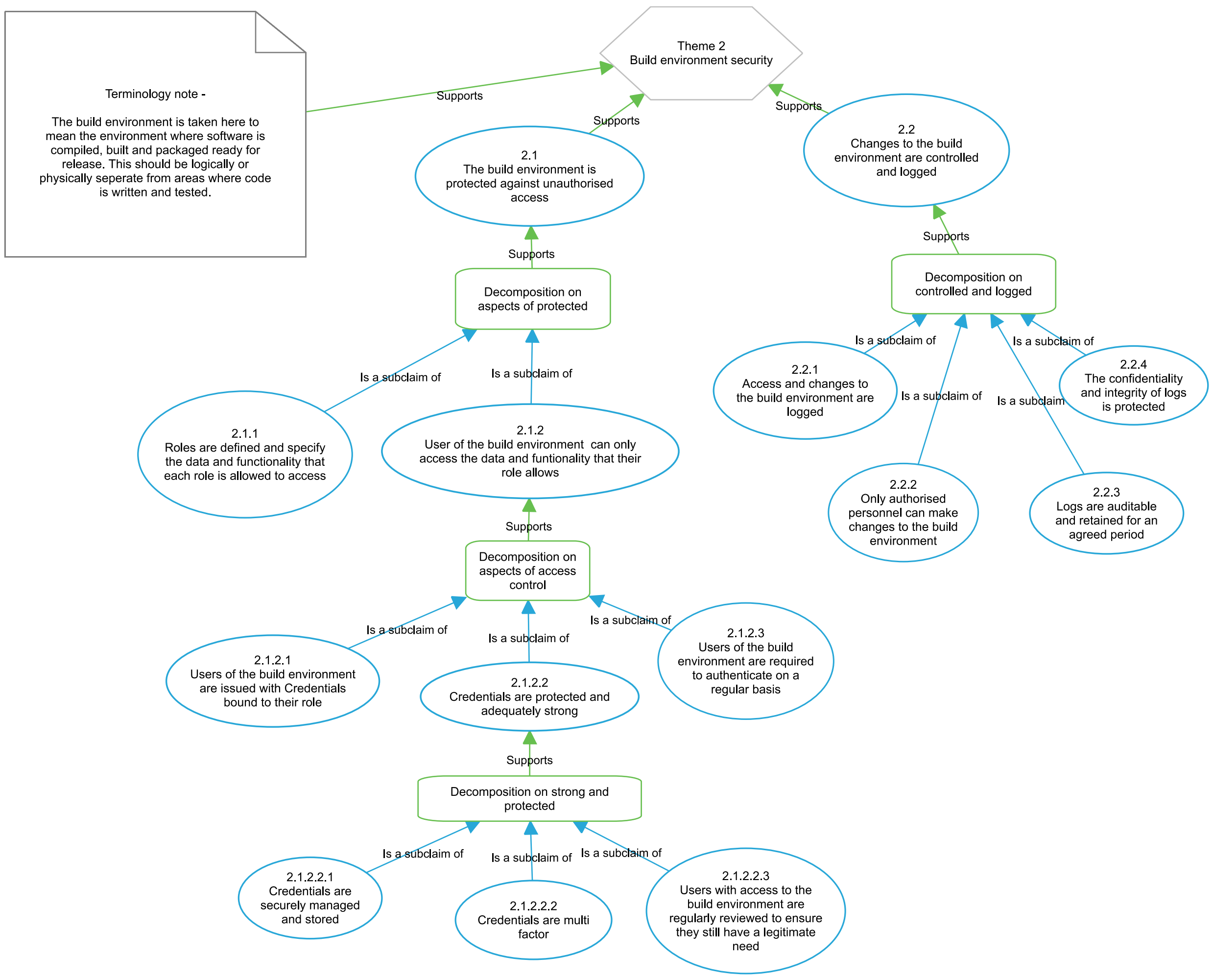
If a vendor is unclear on the intention of one of the claims in this document, then the claims tree can be useful in gaining an understanding of the provenance and rationale for the claim in relation to the principles.

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.

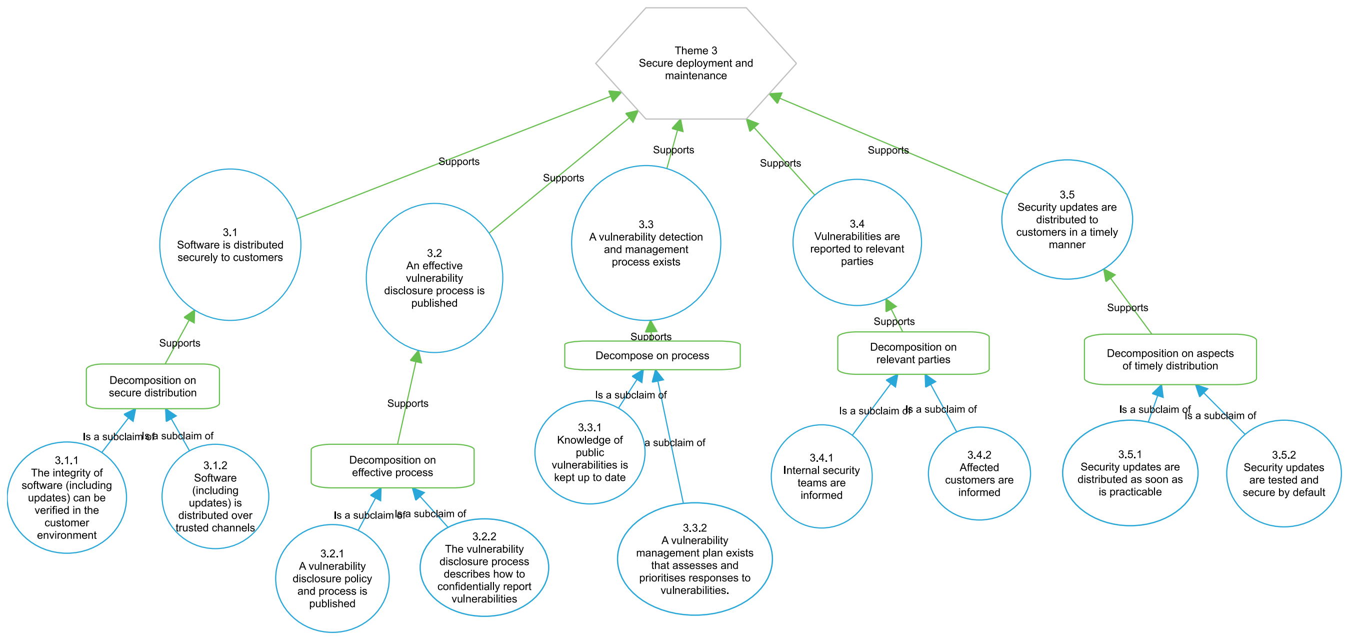
Theme 1: Secure design and deployment


Theme 1 Secure Design and Development
Supports
1.1 An established secure development framework is followed
Decomposition on aspects of established and secure framework
Is a subclaim of
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
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
Substitution
1.2.2.1 Each third party component is secure when first deployed and kept secure throughout its use
1.2.2.1.1 Integrity of third party components and updates is verified
Time split
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
Comments
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

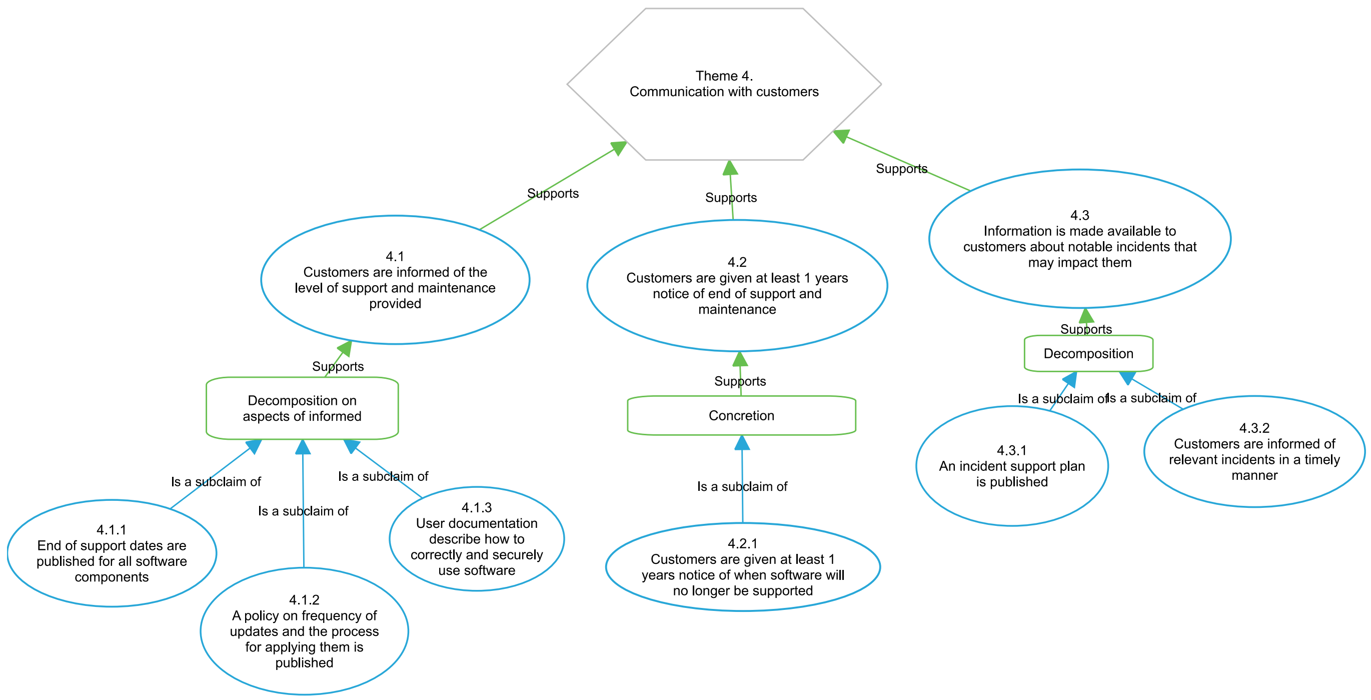
Theme 2: Build environment security


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

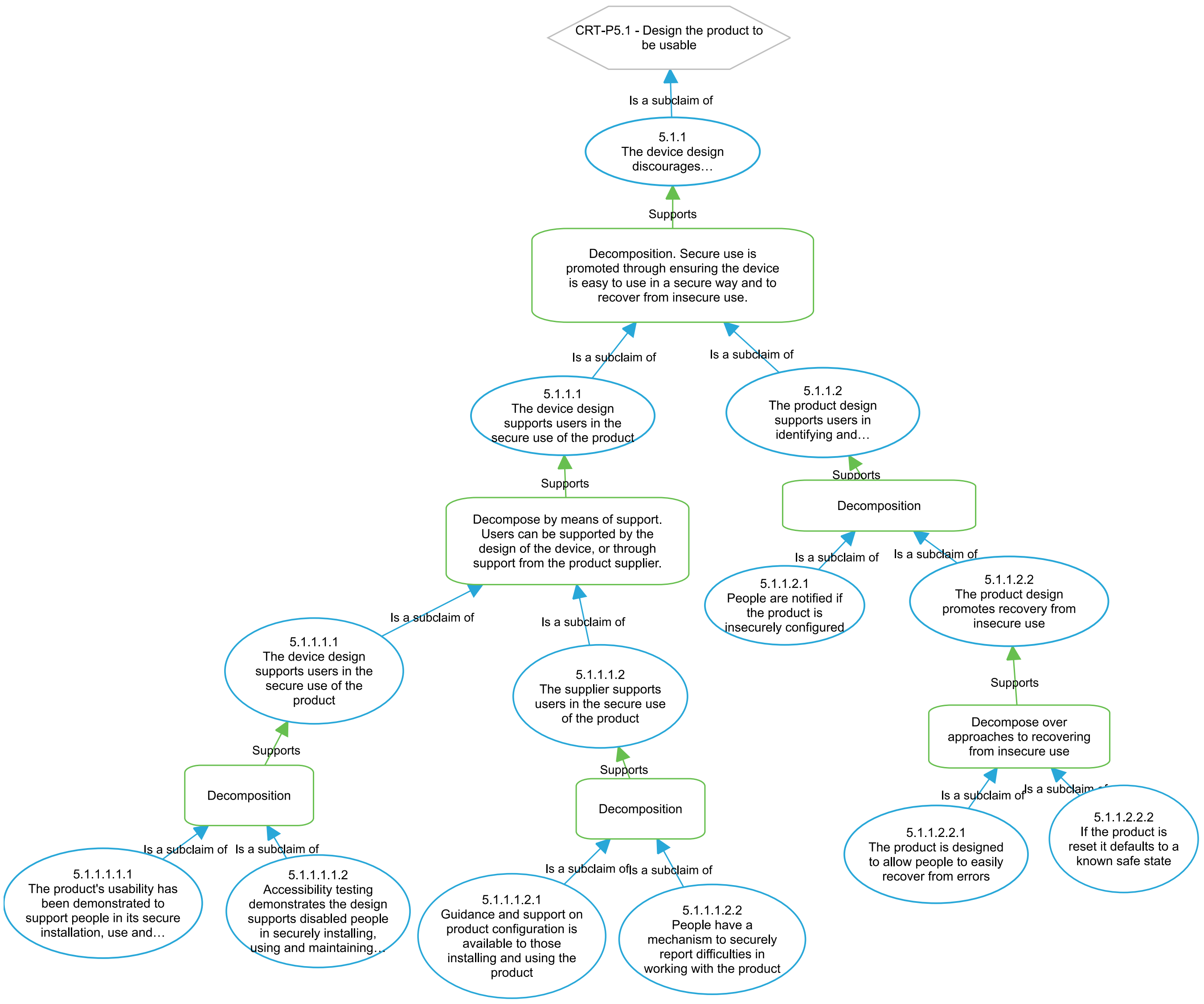
Theme 3: Secure deployment and maintenance

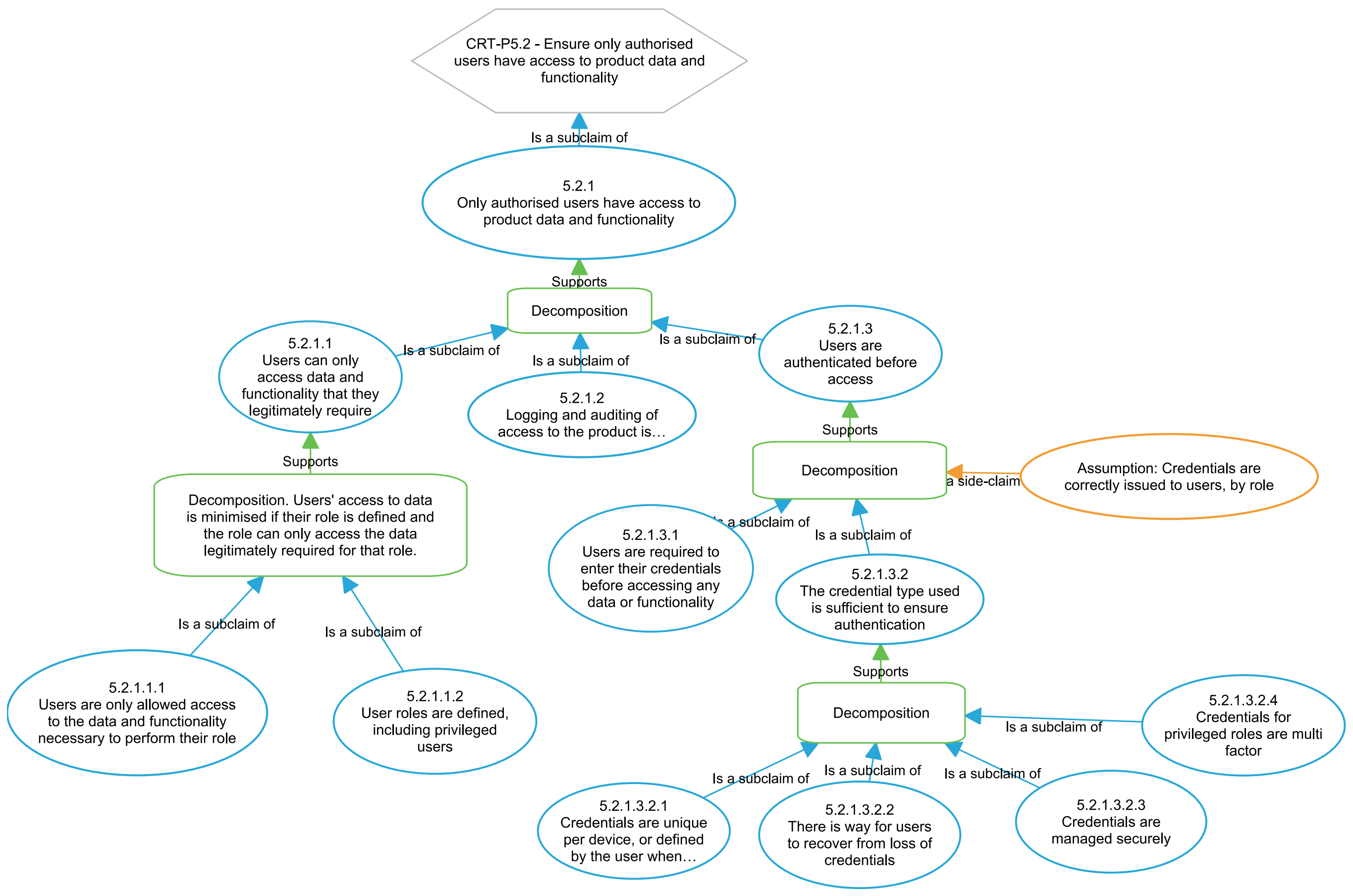

Theme 3
Secure deployment and maintenance
Supports
Supports
Supports
Supports
Supports
3.1
Software is distributed securely to customers
3.2
An effective vulnerability disclosure process is published
3.3
A vulnerability detection and management process exists
3.4
Vulnerabilities are reported to relevant parties
3.5
Security updates are distributed to customers in a timely manner
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
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
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.
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
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 secure by default

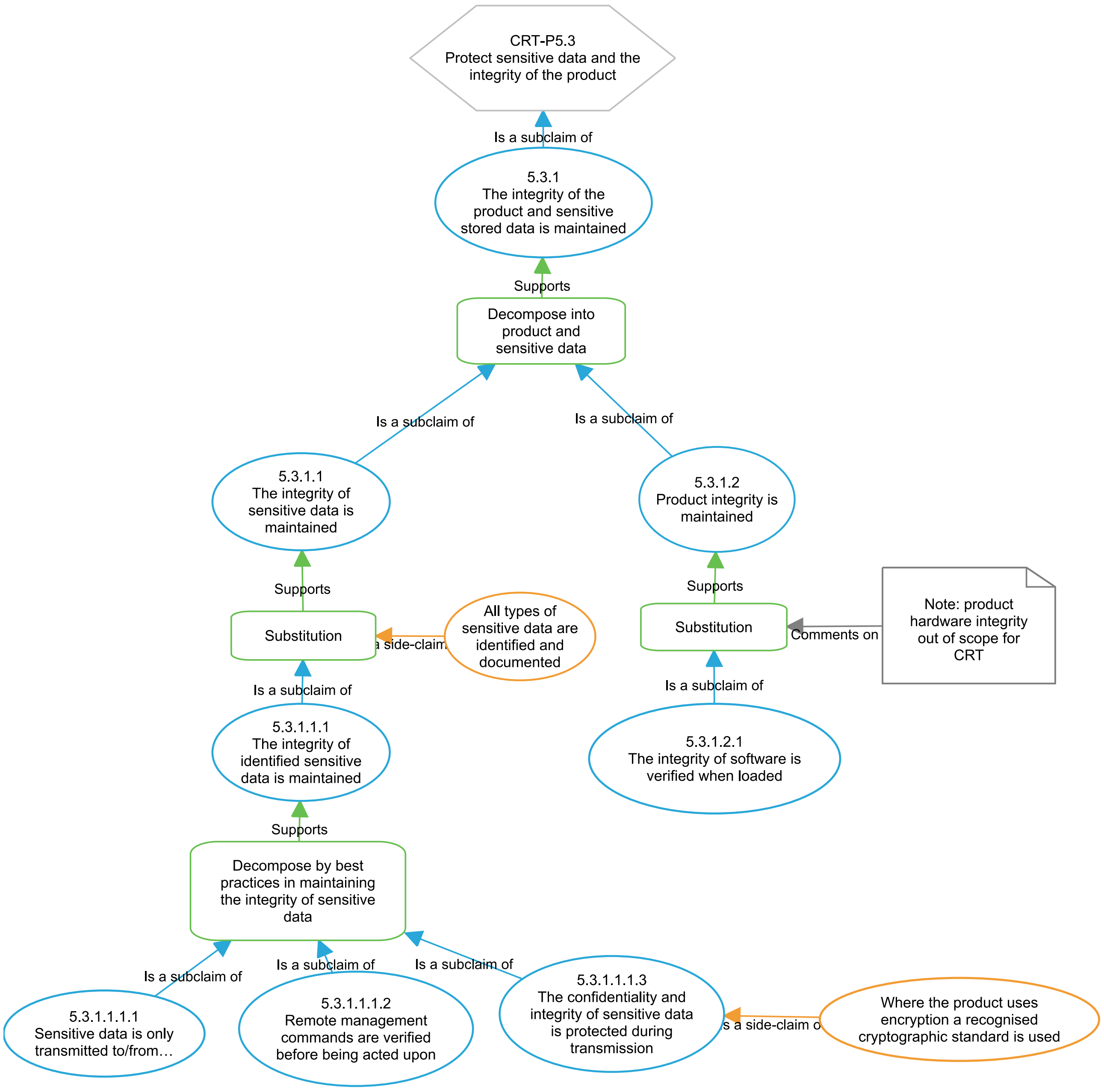
Theme 4: Communication with customers

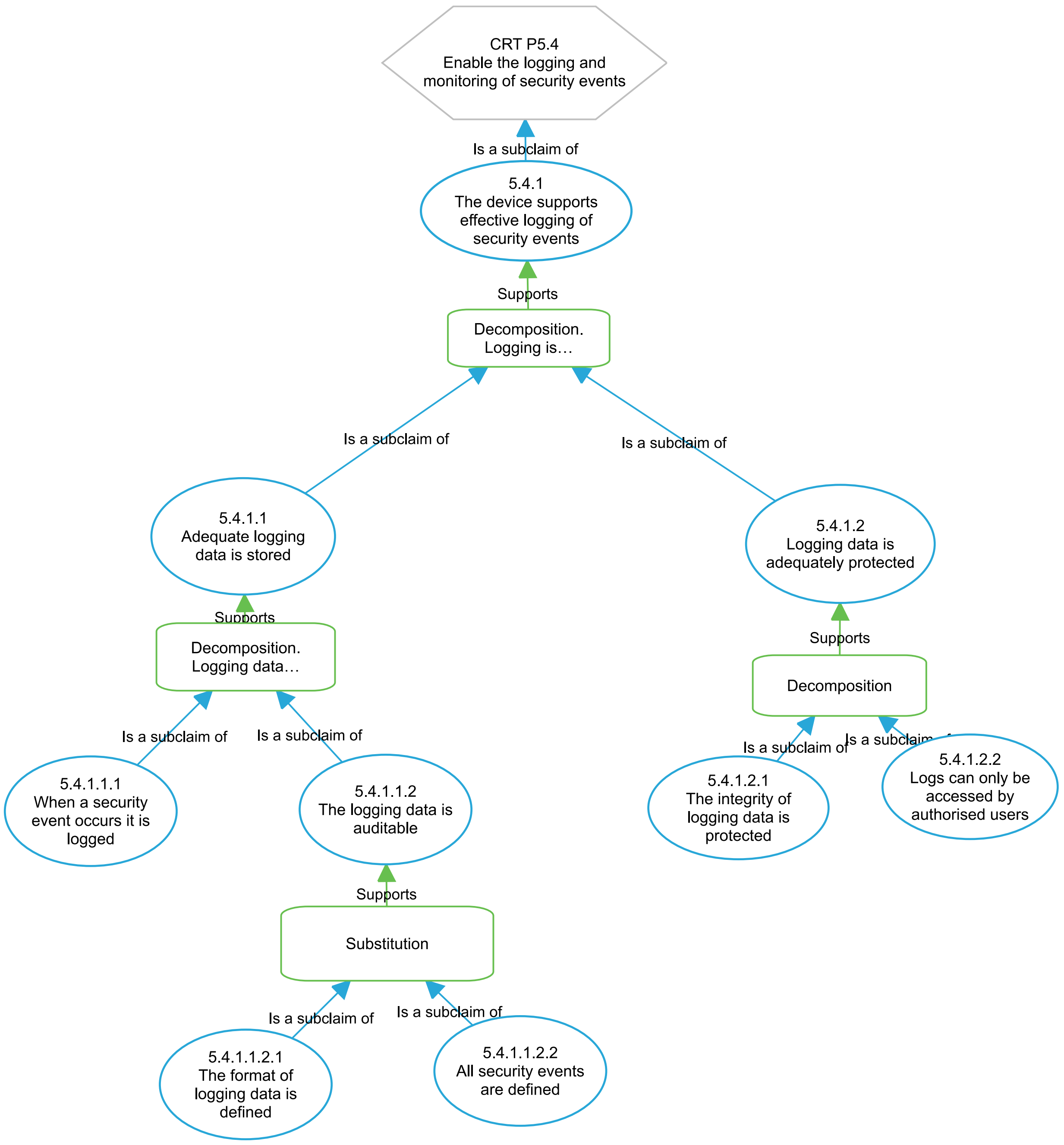

Theme 4.
Communication with customers
Supports
Supports
Supports
4.1
Customers are informed of the level of support and maintenance provided
4.2
Customers are given at least 1 years notice of end of support and maintenance
4.3
Information is made available to customers about notable incidents that may impact them
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
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
Supports
Decomposition
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

Theme 5: Product specific usage, design and operation


CRT-P5.1 - Design the product to be usable
Is a subclaim of
5.1.1 The device design discourages…
Supports
Decomposition. Secure use is promoted through ensuring the device is easy to use in a secure way and to recover from insecure use.
Is a subclaim of
Is a subclaim of
5.1.1.1 The device design supports users in the secure use of the product
5.1.1.2 The product design supports users in identifying and…
Supports
Supports
Decompose by means of support. Users can be supported by the design of the device, or through support from the product supplier.
Decomposition
Is a subclaim of
Is a subclaim of
5.1.1.2.1 People are notified if the product is insecurely configured
5.1.1.2.2 The product design promotes recovery from insecure use
Is a subclaim of
Is a subclaim of
5.1.1.1.1 The device design supports users in the secure use of the product
5.1.1.1.2 The supplier supports users in the secure use of the product
Supports
Decompose over approaches to recovering from insecure use
Supports
Supports
Decomposition
Decomposition
Is a subclaim of
Is a subclaim of
5.1.1.2.2.1 The product is designed to allow people to easily recover from errors
5.1.1.2.2.2 If the product is reset it defaults to a known safe state
Is a subclaim of
Is a subclaim of
Is a subclaim of
Is a subclaim of
5.1.1.1.1.1 The product's usability has been demonstrated to support people in its secure installation, use and…
5.1.1.1.1.2 Accessibility testing demonstrates the design supports disabled people in securely installing, using and maintaining…
5.1.1.1.2.1 Guidance and support on product configuration is available to those installing and using the product
5.1.1.1.2.2 People have a mechanism to securely report difficulties in working with the product


CRT-P5.2 - Ensure only authorised users have access to product data and functionality
Is a subclaim of
5.2.1 Only authorised users have access to product data and functionality
Supports
Decomposition
Is a subclaim of
5.2.1.1 Users can only access data and functionality that they legitimately require
Is a subclaim of
5.2.1.2 Logging and auditing of access to the product is…
Is a subclaim of
5.2.1.3 Users are authenticated before access
Supports
Decomposition. Users' access to data is minimised if their role is defined and the role can only access the data legitimately required for that role.
Is a subclaim of
5.2.1.1.1 Users are only allowed access to the data and functionality necessary to perform their role
Is a subclaim of
5.2.1.1.2 User roles are defined, including privileged users
Supports
Decomposition
a side-claim
Assumption: Credentials are correctly issued to users, by role
Is a subclaim of
5.2.1.3.1 Users are required to enter their credentials before accessing any data or functionality
Is a subclaim of
5.2.1.3.2 The credential type used is sufficient to ensure authentication
Supports
Decomposition
Is a subclaim of
5.2.1.3.2.4 Credentials for privileged roles are multi factor
Is a subclaim of
5.2.1.3.2.1 Credentials are unique per device, or defined by the user when…
Is a subclaim of
5.2.1.3.2.2 There is way for users to recover from loss of credentials
Is a subclaim of
5.2.1.3.2.3 Credentials are managed securely


CRT-P5.3
Protect sensitive data and the integrity of the product
Is a subclaim of
5.3.1
The integrity of the product and sensitive stored data is maintained
Supports
Decompose into product and sensitive data
Is a subclaim of
Is a subclaim of
5.3.1.1
The integrity of sensitive data is maintained
5.3.1.2
Product integrity is maintained
Supports
Supports
Substitution
a side-claim
All types of sensitive data are identified and documented
Substitution
Comments on
Note: product hardware integrity out of scope for CRT
Is a subclaim of
Is a subclaim of
5.3.1.1.1
The integrity of identified sensitive data is maintained
5.3.1.2.1
The integrity of software is verified when loaded
Supports
Decompose by best practices in maintaining the integrity of sensitive data
Is a subclaim of
Is a subclaim of
Is a subclaim of
5.3.1.1.1.1
Sensitive data is only transmitted to/from…
5.3.1.1.1.2
Remote management commands are verified before being acted upon
5.3.1.1.1.3
The confidentiality and integrity of sensitive data is protected during transmission
s a side-claim o
Where the product uses encryption a recognised cryptographic standard is used


CRT P5.4
Enable the logging and monitoring of security events
Is a subclaim of
5.4.1
The device supports effective logging of security events
Supports
Decomposition.
Logging is…
Is a subclaim of
Is a subclaim of
5.4.1.1
Adequate logging data is stored
5.4.1.2
Logging data is adequately protected
Supports
Decomposition.
Logging data…
Supports
Decomposition
Is a subclaim of
Is a subclaim of
Is a subclaim of
Is a subclaim of
5.4.1.1.1
When a security event occurs it is logged
5.4.1.1.2
The logging data is auditable
5.4.1.2.1
The integrity of logging data is protected
5.4.1.2.2
Logs can only be accessed by authorised users
Supports
Substitution
Is a subclaim of
Is a subclaim of
5.4.1.1.2.1
The format of logging data is defined
5.4.1.1.2.2
All security events are defined