CPNI
Centre for the Protection of National Infrastructure

PUBLISH DATE:
June 2021

CLASSIFICATION:
OFFICIAL

CAPSS Evaluation Maintenance – FAQ

CPNI
Centre for the Protection of National Infrastructure

National Cyber Security Centre

Question: I am considering purchasing a CAPSS approved product, but I notice that the latest version of the product was released after the CAPSS approval, and has updated the product functionality. What does this mean for the CAPSS approval status of this latest version?

Answer:

A successful CAPSS Build Standard validation establishes assurance that the developer has processes and practices that are able to maintain the cybersecurity of their CAPSS approved product(s) between evaluation and review periods. The 2-yearly review periods built into the CAPSS lifecycle will include review of the developer's records from some of these processes, including the assessment of changes for any required CAPSS maintenance evaluation activities.

Routine changes and updates to a product can therefore be done by the developer under the approach that was assessed by the Build Standard. But if a change to a CAPSS approved product makes changes to its attack surface (e.g. by adding external interfaces or changing the content or form of those interfaces) and thereby introduces new potential threats then these may need additional independent testing to maintain the CAPSS approval. In this case the developer should have contacted CPNI to confirm any maintenance evaluation activities necessary – if a version of the product is not covered by CAPSS approval because those maintenance evaluation activities have not yet been carried out then the product should be identified as not currently CAPSS approved.

If you are in doubt about the status of a product version, then you can enquire by using the enquiries form on the 'Contact Us' page of the CPNI website.

Question: What changes to a CAPSS approved product should be notified to CPNI, and which would trigger maintenance evaluation work?

Answer:

It is expected that any necessary CAPSS maintenance evaluation work will be done at the time that changes are made by the developer, in order to ensure continuity of the CAPSS approval status for the product. Developers must clearly distinguish product versions that have been identified as requiring CAPSS maintenance evaluation work but where this has not yet been successfully completed.

Some example scenarios are given below, stating in each case whether notification would be required and, if so, the likely extent of maintenance evaluation work.

(a) Developer changes a 3rd party component implementing security in a CAPSS-approved product (e.g. a hardware crypto module or crypto library)

The Build Standard validation carried out during the evaluation of the product includes assessment of the developer's cybersecurity design review process, cybersecurity acceptance tests, and vulnerability handling processes. This gives confidence that changes like this can be made without additional CAPSS evaluator activities between 2-yearly CAPSS

reviews. The exception would be if the change in the 3rd party component changes the external attack surface of the product, in which case CPNI should be notified and will assess the need for additional independent testing of the change. In principle any such testing would be confined to testing any additional attack surface (depending on the extent of the developer's cybersecurity acceptance testing of the 3rd party component, and visibility of its results, some independent testing might be required to confirm that the changes maintain previous levels of assurance).

(b) Developer changes a security mechanism in the product:

A security *mechanism* is part of the implementation of a security *feature*: the mechanism is the way of achieving the feature. So, for example, a particular encryption algorithm, mode and key length, such as AES-128 in CBC mode, would be a mechanism for implementing a data confidentiality feature. If the security mechanism does not change the effect of the existing security feature it implements, then the change does not need to be notified to CPNI. If the security mechanism is specified in the CAPSS SC, then this should be notified to CPNI. In the latter case, and depending on the details of the change, it is likely that a CAPSS lab will need to re-test the new security mechanism (or at least review developer test evidence). In cases where the change occurs near to a 2-year review point, CPNI may agree to defer the independent testing until the review.

Some examples of hypothetical security mechanism changes are:

- Encryption algorithms are specified in DEV.100 (Evaluation/Cryptocheck), so changes to an encryption algorithm should be notified to CPNI. Provided the new mechanism is still approved and has been independently validated then it is unlikely that further CAPSS testing would be needed
- a change in the configuration parameters used by administrators (DEV.301) but that does not change the effect of the security features (e.g. additional grouping or aliases for users or devices) would not need to be notified to CPNI
- a change to the mechanism for time synchronisation that still meets the requirements in DEV.403 would not need to be notified to CPNI
- a change in the protocol used for secure remote management channels (e.g. replacing SSH with TLS): this would affect the cryptographic mechanisms covered by DEV.100 and the scope of DEV.406 (Encrypt communications traffic over untrusted link) and DEV.505 (Remote management authentication), and therefore should be notified to CPNI. Some review of the developer's use of channel parameters (e.g. protocol version, authentication methods configured) and configuration according to relevant security guidance (e.g. NCSC guidance on TLS configuration) would be needed, possibly with some additional testing to confirm the secure configuration.

(c) Developer implements a new security feature in the product:

If the new security feature is related to the CAPSS SC (i.e. if it implements one of the SC requirements, or provides a way to relax one of those requirements, or an alternative function that is subject to SC requirements) then this is treated the same as example (b) above of changing a security mechanism.

Some examples of hypothetical security feature changes are:

- adding multi-factor authentication to a component that previously used only passwords: this maintains the product's ability to meet DEV.502 and therefore would not have to be notified to CPNI
- adding an additional remote access capability for a component: this would change the product's attack surface and the new interface would need to be added to the scope of the authentication management requirements in DEV.500-505. This change should therefore be notified to CPNI and would require additional analysis of the remote access design and testing of the channel configuration.

(d) Developer adds a new interface (e.g. remote management over SSH/TLS/IPsec, or a new protocol (logical interface) over an existing physical interface):

The specific case of adding a remote management interface was described as an example in (c) above. More generally for cases of this type, it is likely that a CAPSS lab will need to independently examine and re-test the security of the new interface: the testing in this case is likely to be more extensive than for cases of additional testing in (b) above because the new interface will require new design information to be reviewed, new tests to be created (probably with reference to design information) and new fuzz testing for DEV.407 & VER.407. If the new interface is wireless or introduces cloud services, then further additional analysis and testing will need to be carried out by the CAPSS lab for the relevant requirements (e.g. DEV.401 and DEV.700).

(e) Developer changes their own security-implementing hardware or software (other than for changes above) but maintains the same specification of security requirements

This is a case where the Build Standard validation establishes that the developer's internal security review and test processes are sufficient to maintain assurance between CAPSS reviews. Changes that maintain the same specification benefit from the previous understanding and test scope and hence support confidence in the continuing assurance.

(f) Developer adds new non-security functionality

Adding non-security functionality covers cases such as extending the number of sensors that a product can communicate with, adding overlays to input feeds or camera tracking control in a video management system, or adding new operator workflows. This is another case where the Build Standard validation establishes that the developer's internal security review and test processes are sufficient to maintain assurance between CAPSS reviews – provided that the new functionality is not one of the cases (b) - (d) above.

(g) Developer fixes a bug

This is another case where the Build Standard validation establishes that the developer's internal security review and test processes are sufficient to maintain assurance between CAPSS reviews, again assuming that the bug fix does not overlap with (b) - (d).