HCD0006 - Technical issue in Tests assurace activity for FCS_CKM.4 Cryptographic key destruction

Version: 1, Published: 2024-02-04

Impacted Documents

CPP_HCD_V1.0_supporting_doc

References

FCS_CKM.4 Cryptographic key destruction

Issue Description

The Tests assurance activity contains a paragraph that originates from CPP_FDE_AA_V2.0E_supporting_doc. Due to the differences between Full Drive Encryption Authorization Acquisition technology type and Hardcopy Devices technology type, this paragraph cannot be applied to Hardcopy Devices.

Resolution

Delete the paragraph originating from CPP_FDE_AA_V2.0E_supporting_doc from the Tests assurance activity.

CPP_HCD_V1.0_supporting_doc

The SD is updated as follows (yellow highlights for additions, strikethrough for deletions) per section that is being updated:

2.2.4.4. Tests

For these tests the evaluator shall utilize appropriate development environment (e.g., a Virtual Machine) and development tools (debuggers, simulators, etc.) to test that keys are cleared, including all copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

Test 1 [conditional]: Applied to each key held as in volatile memory and subject to destruction by overwrite by the TOE (whether or not the value is subsequently encrypted for storage in volatile or nonvolatile memory). In the case where the only selection made for the key destruction method

was removal of power, destruction of reference to the key directly followed by a request for garbage collection, or memory management, then this test is unnecessary. The evaluator shall:

1. Record the value of the key in the TOE subject to clearing.
2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1.
3. Cause the TOE to clear the key.
4. Cause the TOE to stop the execution but not exit.
5. Cause the TOE to dump the entire memory of the TOE into a binary file.
6. Search the content of the binary file created in Step #5 for instances of the known key value from Step #1.

Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.

~~The following tests apply only to selection volatile memory), since the TOE in this instance has more visibility into what is happening within the underlying platform (e.g., a logical view of the media). In selection non-volatile storage), the TOE has no visibility into the inner workings and completely relies on the underlying platform, so there is no reason to test the TOE beyond test 1. For selection volatile memory), the following tests are used to determine the TOE is able to request the platform to overwrite the key with a TOE supplied pattern.~~

Tracking

Issue #19