
I-0350: Clarification Of Resources/Objects For Residual Information Protection

TYPE: NIAP Interpretation
NUMBER: I-0350
STATUS: Ready for External Repost after Interpretations Board Rework/Review

TITLE: Clarification Of Resources/Objects For Residual Information Protection

SOURCE REFERENCE: CC v2.1 Part 2 Subclause 6.9 FDP_RIP
CC v2.1 Part 2 Subclause F.9 FDP_RIP

RELATED TO:
[I-0041](#) Object Reuse Applies To All System Resources
[I-0356](#) FDP_RIP Annex: Reuse Of Subject Data Notes

ISSUE:

The focus of the Residual information protection (FDP_RIP) family is not clearly stated in the Common Criteria. The focus of residual information protection is preventing leakage of information from one instantiation of a type of object to another instantiation of that type of object. This should include cleansing of the TSF-internal data structures that are used to construct objects and are visible through the object. However, this aspect doesn't come across clearly in the current words.

STATEMENT

Residual information protection applies to those TSF-internal structures that are visible though the TSF-interface and are used to implement the object for whom residual information protection applies.

SPECIFIC INTERPRETATION

To address this interpretation, the following changes are made to CC v2.1, Part 2 (additions marked thusly; deletions marked ~~thusly~~):

- In Subclause 6.9, paragraph 223 is replaced with the following:

This family addresses the need to ensure that information contained in a resource is not available when resources are deallocated from one object and reallocated to a different object ~~deleted information is no longer accessible, and that newly created objects do not contain information that should not be accessible.~~ The term "resources" refers to those TSF-internal structures that are visible though the TSF-interface and are used to implement the object for whom residual information protection applies. This family requires protection for information that has been logically deleted or released, but may still be present within the TOE.

- In Subclause F.9, paragraph 887 is replaced with the following:

This family addresses the need to ensure that information contained in a resource is not available when resources are deallocated from one object and reallocated to a different object ~~deleted information is no longer accessible, and that newly created objects do not contain information from previously used objects within the TOE.~~ The term "resources" refers to those TSF-internal structures that are visible though the TSF-interface and are used to implement the object for whom residual information protection applies. This family addresses all entities meeting the definition of **object** as provided in Part 1, Clause 2 (i.e., entities that contain or receive information, and are acted upon by subjects), but does not address objects stored off-line. Note that the definition of **object** is broad enough to include processes if they may be acted upon by subjects, contain or receive information, and the resources that constitute the process are reused as processes are created and destroyed.

PROJECTED IMPACT:

Negligible impact anticipated.

SUPPORT:

This interpretation clarifies the meaning of resource as used in the FDP_RIP components. It clarifies that the focus of RIP is those portions of the structures that are externally visible.

Additionally, this interpretation provides clarification with respect to the applicability of RIP to "subjects", which are typically processes. Of particular concern are the TSF-structures used to construct processes; the goal is to ensure that any externally-visible information is cleansed so that new processes start out "clean".

Originally, it was felt that new components were required to address processes. Further analysis shows that processes typically meet the broad Common Criteria definition of object: they contain or receive information, and are acted upon by subjects (i.e., created, destroyed, observed, etc.). This interpretation clarifies that the broad definition of object covers processes.

Note that process/subject residual information is also addressed through FPT_SEP. The distinction is that FDP_RIP addresses process creation and destruction. FPT_SEP, on the other hand, addresses reuse of externally visible structures as processes are chosen and removed from execution (i.e., swapped in/out).