

Assurance Considerations for a Highly Robust TOE

Thuy D. Nguyen, Cynthia E. Irvine, Timothy E. Levin, Michael A. McEvelley

Abstract

The U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness (SKPP) is undergoing evaluation. During its authoring process, new extended functional and assurance requirements were introduced to resolve assurance issues associated with TOE hardware, trusted initialization and trusted recovery.

For high robustness, domain separation and self-protection are architectural assurances only realistically achieved with hardware support. Since the Common Criteria does not include requirements for establishing assurance in security-relevant hardware mechanisms, a new Platform Assurance class was introduced. It provides a framework to determine the security relevance of commercially-available hardware based on its interfaces to software and to establish trust in those hardware mechanisms deemed security-relevant.

Requirements for TOE initialization behavior and for establishing trust in that behavior are not prescribed in the CC Version 2.3. Although CC Version 3.1 does define secure initialization assurances, they are not sufficient for high robustness. A new Trusted Initialization assurance family was introduced to require a TOE initialization function that reliably establishes the TSF in an initial secure state, verifies TSF integrity during initialization, handles failures during initialization, does not arbitrarily interact with the TSF following TOE initialization, provides self-protection during initialization, and addresses the threat that the TSF is initialized by other components executing on the TOE.

Existing trusted recovery requirements emphasize the means of failure handling (i.e., manual versus automated) instead of protecting against further compromise during a recovery from an insecure state to a secure state. Extended trusted recovery requirements were introduced to require the TSF to attempt self-recovery to a secure state when the TSF detects that it is in an insecure state. To avoid ambiguity, the TOE developer must enumerate pair-wise recovery conditions and their associated actions and provide appropriate evidence that secure state results from the identified action.