National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme
Validation Report

U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness

Report Number: CCEVS-VR-07-0055
Dated: 17 July 2007
Version: 1.0

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6740
Fort George G. Meade, MD 20755-6740

ACKNOWLEDGEMENTS

Validation Team

Franklin Haskell
The MITRE Corporation
Bedford, Massachusetts

James Donndelinger
The Aerospace Corporation
Columbia, Maryland

Common Criteria Testing Laboratory

COACT CAFÉ Laboratory
Columbia, Maryland

Table of Contents

List of Tables

1 Executive Summary

The evaluation was carried out in accordance with the Common Criteria Evaluation and Validation Scheme (CCEVS) process and scheme and was completed during April, 2007. The criteria against which the Separation Kernel Protection Profile (SKPP) was judged is described in the Common Criteria for Information Technology Security Evaluation, Version 2.3 and International Interpretations effective on July 1, 2004. The evaluation methodology used by the COACT CAFÉ Lab evaluation team to conduct the evaluation is the Common Methodology for Information Technology Security Evaluation, Version 2.3, Part 3, Class APE: Protection Profile Evaluation.

The assurance activities in this CC class offer confidence that the SKPP contains requirements that are:

1. justifiably included to counter stated threats and meet realistic security objectives,
2. internally consistent and coherent and
3. technically sound.

COACT, the Common Criteria Testing Laboratory, is certified by the NIAP validation body for laboratory accreditation. The CCTL has presented CEM work units and rationale that are consistent with the CC, the CEM and CCEVS publication number 4 Guidance to CCEVS Approved Common Criteria Testing Laboratories. The CCTL team concluded that the requirements of the APE class have been met. Therefore, a pass verdict has been issued by the CCTL for the protection profile assurance family. The validation team followed the procedures outlined in the Common Criteria Evaluation Scheme publication number 3 for Technical Oversight and Validation Procedures. The validation team has observed that the evaluation and all of its activities were in accordance with the Common Criteria, the Common Evaluation Methodology, and CCEVS policy. The validation team concludes that the evaluation has completed and the evaluation team's results are valid. Therefore, the Common Criteria Evaluation and Validation Scheme grants a Common Criteria Certificate to the sponsor, acknowledging the successful completion of the evaluation and the validity of this Common Criteria Protection Profile.

1.1 Evaluation Details

Table 1 – Evaluation Details

Evaluated Product:	U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness Version 1.03, June 29, 2007
Registration:	Information Assurance Directorate
Keywords:	separation kernel, high robustness, data isolation, information flow control, partition, cryptography, commercial-off-the-shelf (COTS)
Developer:	Information Assurance Directorate, National Security Agency, 9800 Savage Road, Fort George G. Meade, MD

20755-6000

CCTL: COACT, Inc., Rivers Ninety Five, 9140 Guilford Road, Suite G, Columbia, MD 21046-2587

Kickoff Date: December 2006

Completion Date: July 17 2007

CC: Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005.

Interpretations: I-407

CEM: Common Evaluation Methodology for Information Technology Security, Part 1: Introduction and General Model, Version 0.6, January 1997; Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology, Version 2.3, August 2005.

Description: The SKPP specifies the security functional and assurance requirements for the separation kernel portion of an operating system. Unlike those traditional security kernels which perform all trusted functions for a secure operating system, a separation kernel's primary security function is to partition (viz. separate) the subjects and resources of a system and to enforce the rules for authorized information flows between those partitions as defined by the security policy. .

Evaluation Personnel:
COACT:
Brian Pleffner
Greg Beaver

Validation Team:
MITRE:
Franklin Haskell
Aerospace:
James Donndelinger

1.2 Interpretations

Interpretation ID	Impact on CC Requirements	Impact on CEM Work Units	Comment
I-407	FAU_GEN.1	None	Applied

1.3 Protection Profile Summary

The SKPP contains requirements which specify the security functional and assurance requirements for the separation kernel portion of an operating system. Compliant TOE implementations provide a highly robust foundation for system services and applications in mission-critical embedded systems, and a high degree of assurance for the enforcement of related security policies. They do this by providing facilities that *partition* or separate the various pieces of software built to operate in this environment; and mechanisms that strictly control the flow of information between the partitions created.

A TOE conforming to the SKPP requirements will include the following security features:

1. Trusted Delivery: using cryptographically-based techniques.
2. Configuration: to translate human-readable representations into OS useable data.
3. Load: to copy configuration data onto the execution platform.
4. Initialization: to copy the software with its configuration data onto the execution platform and start it for application operation.
5. Information flow control that enforces strict partition isolation, with the exception of explicit interactions specified by the configuration data
6. Cryptographic mechanisms that provide functions to verify the integrity of TSF code and data during trusted delivery
7. Trusted initialization and recovery functions
8. Detection and response to security function failures
9. Generation of audit data

These facilities create a software execution environment that is suitable for applications requiring high assurance and robustness.

2 TOE Security Environment

2.1 Threats to Security

The following are the threats that the evaluated product addresses:

Table 2 – Threats

Threat Identifier	Threat Description
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE (including the misapplication of the protections afforded by the PIFP), or install a corrupted TOE resulting in ineffective security mechanisms.
T.ALTERED_DELIVERY	The TOE may be corrupted or otherwise modified during delivery such that the on-site version does not match the master distribution version.
T.CONFIGURATION_CHANGE	The lack of TSF-enforced constraints on the ability of an authorized subject to invoke or dictate how the TOE is reconfigured may result in the TOE transitioning to an insecure (unknown, inconsistent, etc) state.
T.CONFIGURATION_INTEGRITY	The TOE may be placed in a configuration that is not consistent with that of the configuration vector due to

	the improper loading of the configuration vector or incorrect use of the configuration vector during TOE initialization.
T.COVERT_CHANNEL_EXPLOIT	An unauthorized information flow may occur between partitions as a result of covert channel exploitation.
T.DENIAL_OF_SERVICE	A malicious subject may block others from system resources (e.g., system memory, persistent storage, and processing time) via a resource exhaustion attack.
T.INCORRECT_CONFIG	The configuration vectors are not an accurate and complete description of the operational configuration of the TOE as used by an organization.
T.INCORRECT_LOAD	The software portion of the TSF implementation and/or configuration vectors are not correctly converted into a TOE-useable form.
T.INSECURE_STATE	The TOE may be placed in an insecure state as a result of an erroneous initialization, halt, reconfiguration or restart, transition to maintenance mode, or as a result of an unsuccessful recovery from a system failure or discontinuity.
T.LEAST_PRIVILEGE	The design and implementation of the TSF internals may not suffice to limit the damage resulting from accident, error or unauthorized use.
T.POOR_DESIGN	Unintentional or intentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious subject.
T.POOR_IMPLEMENTATION	Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious subject.
T.POOR_TEST	Lack of or insufficient evaluation and runtime tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered.
T.TSF_COMPROMISE	A malicious subject may cause TSF data or executable code to be inappropriately accessed (viewed, modified, executed, or deleted).
T.UNAUTHORIZED_ACCESS	A subject may gain access to resources or TOE security management functions for which it is not authorized according to the TOE security policy.

2.2 Security Policy

Table 3 – Policies

Policy Identifier	Policy Description
P.ACCOUNTABILITY	The TOE shall provide the capability to make available information regarding the occurrence of security relevant events.
P.CONFIGURATION_CHANGE	The TOE shall support the capability to perform a static configuration change. The TOE may also provide the capability for an authorized subject to select or redefine the configuration vector to be used upon TOE startup, TOE restart or TOE reconfiguration.

P.CRYPTOGRAPHY	The TOE shall use NSA approved cryptographic mechanisms.
P.INDEPENDENT_TESTING	The TOE shall undergo independent testing.
P.RATINGS_MAINTENANCE	A plan for procedures and processes to maintain the TOE's rating shall be in place to maintain the TOE's rating once it is evaluated.
P.SYSTEM_INTEGRITY	The TOE shall provide the ability to periodically validate its correct operation.
P.USER_GUIDANCE	The TOE shall provide documentation regarding the correct use of the TOE security features.
P.VULNERABILITY_ANALYSIS_AND_ TEST	The TOE shall undergo independent vulnerability analysis and penetration testing by NSA to demonstrate that the TOE is resistant to an attacker possessing a high attack potential.

2.3 Security Usage Assumptions

The specific conditions below are assumed to exist in a PP-compliant TOE environment:

Table 4 – Security Usage Assumptions

Assumption Identifier	Assumption Description
A.PHYSICAL	It is assumed that the non-IT environment provides the TOE with appropriate physical security commensurate with the value of the IT assets protected by the TOE.
A.SUBJECT_ALLOCATION	It is assumed that a trusted individual will create configuration vectors such that, for those partitions to which subjects are allocated, each partition is allocated one or more subjects (i.e., subjects with homogeneous access requirements, or subjects with heterogeneous access requirements) that are appropriate for the policy abstraction supported by the TOE.
A.COVERT_CHANNELS	If the TOE has covert storage and/or timing channels, then for all subjects executing on that TOE, it is assumed that relative to the IT assets to which they have access, those subjects will have assurance sufficient to outweigh the risk that they will violate the security policy of the TOE by using those covert channels.
A.TRUSTED_FLOWS	For any subject configured to have unrestricted access in multiple policy equivalence classes, it is assumed that the subject is trusted at least with assurance commensurate with the value of the IT assets in all equivalence classes to which it has access

2.4 Clarification of Scope

Any TOE under evaluation which claims conformance to the SKPP will include hardware as part of its evaluation as it's required for all medium and high robustness evaluations.

There are parts of the TOE that are not part of the TSF for which specific assurances are required. These include facilities used to configure and start (in the very general sense) the TOE. In particular creation of the partitions and the database containing the security policy, which is to say the allowable data flows from partition to partition, are included. A number of requirements are levied upon these specifically because a great deal of trust is placed on these for correct operation of the TOE.

3 Security Content of the Protection Profile

An SKPP-compliant TOE will provide the following features:

1. Information flow control that enforces strict partition isolation, with the exception of explicit interactions specified by the configuration data
2. Cryptographic mechanisms that provide functions to verify the integrity of TSF code and data during trusted delivery
3. Trusted initialization and recovery functions
4. Detection and response to security function failures
5. Generation of audit data
6. Trusted Delivery: using cryptographically-based techniques.
7. Configuration: to translate human-readable representations into OS useable data.
8. Load: to copy configuration data onto the execution platform.
9. Initialization: to copy the software with its configuration data onto the execution platform and start it for application operation.

4 Results of the Evaluation

The Common Criteria Testing Laboratory team conducted the evaluation according to the CC and the CEM and concluded that the requirements of the APE class were met. Therefore, a pass verdict has been issued for the protection profile assurance family.

5 Validator Comments/Recommendations

The validation team believes that the SKPP is a very good set of requirements for particular types of high assurance applications. Such applications would be rigorously defined and implemented. The SKPP is not intended for general purpose operating systems though such systems could be installed “on top of” a separation kernel but such use of a separation kernel would not be readily apparent to the users of the general purpose OS.

Three new families of explicit requirements have been created whose requirements are virtual duplicates of requirements in other existing CC requirement families: ADV_CTD, ADV_INI, and ADV_LTD. This was done because these are requirements levied upon what might be called “support functions” rather than “security functions”. Examination of the explicit requirements will reveal that they are closely related to other ADV, ATE families and that therefore the methodology specified in the CEM for their evaluation can be transplanted onto the new families.

Audit requirements are present but minimal in scope. It is up to the application designers to decide how much audit capability is necessary beyond the requirements; i.e. audit trail maintenance, automatic actions, reduction of audit data and so forth are not included. In certain real-time applications these functions are not even considered for inclusion in the system. Other systems could well include them but make their operation secondary to other tasks of the applications. The audit requirements present were created by rewriting several of the CC audit requirements as explicit requirements with the effect of limiting their functionality. This is entirely appropriate.

6 Glossary

No definitions beyond those in the CC or CEM are supplied.

7 Bibliography

The Validation Team used the following documents to produce this Validation Report:

[1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, Version 2.2, January 2004, CCIMB-2004-01-001.

[2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, Version 2.2, January 2004, CCIMB-2004-01-002.

[3] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, Version 2.2, January 2004, CCIMB-2004-01-003.

[4] Common Methodology for Information Technology Security: Evaluation Methodology, Version 2.2, January 2004, CCIMB-2004-01-004.

[5] NIAP Common Criteria Evaluation and Validation Scheme for IT Security, Guidance to Common Criteria Testing Laboratories, Version 1.0, March 20, 2001.