Copy No: _____

Review of the Composability Problem for System Evaluation

Dan Craigen
ORA Canada

Mark Saaltink
ORA Canada

ORA Canada

Contractor's Report

DRDC Ottawa CR 2004-196

November 2004

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2004		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Review of the Composability Problem for System Evaluation (U)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defence R&D Canada -Ottawa,3701 Carling Ave,Ottawa Ontario,CA,K1A 0Z4**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
This report presents the results of a review of the security evaluation of composite systems to determine the current status of this topic, especially with respect to work that has been done in this area and tools and techniques that have been developed. Recommendations on how security evaluation could be applied to the DRDC/NIO Secure Access Manager are also included.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**53**	

Abstract

This report presents the results of a review of the security evaluation of composite systems to determine the current status of this topic, especially with respect to work that has been done in this area and tools and techniques that have been developed. Recommendations on how security evaluation could be applied to the DRDC/NIO Secure Access Manager are also included.

This page intentionally left blank.

Executive summary

Defence R&D Canada - Ottawa (DRDC Ottawa) is conducting research into secure role-based access management and the application of commercial implementations of Public Key Infrastructure (PKI) and Privilege Management Infrastructure (PMI) technologies to provide this capability. A key aspect of this research is the integration of multiple commercial products into proof-of-concept demonstration systems. Feedback from the demonstrators has identified the evaluation and accreditation of such a composite system as an important issue that requires further research.

The requirement for this project is for a review of the security evaluation of composite systems to determine the current status of this topic, especially with respect to work that has been done in this area and tools and techniques that have been developed.

We conclude that while there have been successes in evaluating composable systems, there are numerous issues and trouble spots, some of which are discussed below. Evaluating composite systems, even at the lower levels of the Common Criteria, appears to be costly, performed by large, well-funded organizations and requiring heroic effort. Even though some complex systems have been evaluated up to EAL4+ (e.g. Windows 2000 Professional, Server and Advanced Server with SP3 with Q326886), there is a sense that the approaches do not scale in a satisfactory manner. If they did scale or were cost effective the list of evaluated products would be of much greater magnitude.

There is a significant gap between the tools, techniques and requirements used in practice and emerging results. In fact, our opinion is that most research work focuses on the Common Criteria development component; the massive issues pertaining to documentation and traceability are hardly touched.

It is readily apparent that general commercial practice in system development is at odds with evaluation practices (at least, to some extent). The pressures of the market place, the need to make rapid changes, the imperative of time to market, the need for broad functionality over assurance all trump technical concerns for assurance. Yet, in our opinion, there is value to be derived, even in a purely commercial world, from some aspects of the evaluation process. Protection Profiles, in particular, are excellent documents (if done right) for identifying the security environment, security objectives, security functional requirements and security assurance requirements.

We conclude that current evaluation techniques are generally inadequate.

Despite the difficulties in performing evaluations of composite systems, there is hope that the Secure Access Manager (SAM) could be evaluated to EAL2 or EAL3. These levels avoid the need to delve into the implementation details of third-party components.

Craigen, D.Saaltink, M. 2004. Review of the Composability Problem for System Evaluation. DRDC Ottawa CR-2004-196

Table of contents

List of tables

Acknowledgements

Dan Craigen would like to thank Steve Zeber for his input during the project. Mr. Craigen also thanks his colleague Mark Saaltink for providing support to his efforts.

This page intentionally left blank.

1. Introduction

Defence R&D Canada - Ottawa (DRDC Ottawa) is conducting research into secure role-based access management and the application of commercial implementations of Public Key Infrastructure (PKI) and Privilege Management Infrastructure (PMI) technologies to provide this capability. A key aspect of this research is the integration of multiple commercial products into proof-of-concept demonstration systems. Feedback from the demonstrators has identified the evaluation and accreditation of such a composite system as an important issue that requires further research.

1.1 Requirement

The requirement for this project is for a review of the security evaluation of composite systems to determine the current status of this topic, especially with respect to work that has been done in this area and tools and techniques that have been developed.

1.2 Tasks to be Performed

Through this project, the contractor performed the following tasks:

1. Determined the scope of the problem and identified the basic terms and concepts.

2. Conducted a general survey of literature published on the topic.

3. Investigated, and documented, insofar, as information is available, the views of the Canadian Security Establishment (CSE) and the US National Security Agency (NSA) and any work they have done on this topic.

4. Investigated, and documented, insofar, as information is available, the views of, and any work that may have been done in, other NATO nations, particularly the UK.

5. Investigated and documented the approach of the current US Content-based Information Security (CBIS) project to the evaluation of a composite system.

6. Noted those cases in which information could not be provided to the investigation because of classification or other access restrictions.

7. Recommended, based on the information collected in Tasks 1 – 3, a direction for further work in this area.

1.3 Deliverables

The contractor delivered the following:

1. Within two weeks of contract award, a draft one-level table of contents for the final report and an outline of the proposed approach.

2. Complete drafts of the final report not later than one week prior to the contract termination date.

3. A final report in electronic Microsoft Word format (one copy) and printed paper format (two copies, bound according to DRDC instructions).

1.4 Context

Software systems are seldom built from scratch. This trend began with the first software libraries and run-time systems, and continues with component-based architectures and distributed systems. A number of frameworks, such as Common Object Request Broker Architecture (CORBA), Java Beans, Component Object Model (COM), and the Distributed Component Object Model (DCOM), provide mechanisms for combining software components into a system. Applications are routinely built using existing products to provide directory services (e.g., the Lightweight Directory Access Protocol (LDAP)), authentication (e.g., Kerberos), databases, user interface (e.g., using Hyper-Text Markup Language (HTML) and Javascript which is interpreted by a browser), and so on. The Secure Access Manager SAM) prototype [23,24,41] exemplifies this type of development, by its use of commercial components and systems for authentication, access management, directory services, and data storage.

Using existing components in system construction may have several advantages:

- development time may be reduced, as the existing components can provide sophisticated services,

- quality may be increased, as the existing components are likely to be more mature and better debugged than newly written code,

- interoperability with other software may be improved, for example, by sharing a common repository, and

- management of the new system may be simpler if the existing components have familiar management interfaces.

Reduced development time and costs are key motivators.

Software evaluation is important in many application areas. For government and military applications, the Common Criteria [1] is the most important standard for security evaluation. The Common Criteria impose requirements on the development and documentation of systems. These requirements are not always easily satisfied in a component-based system. Components are often Commercial-Off-The-Shelf (COTS) that are protected, with the implementation details kept secret. It may even be a violation of the component's license agreement to apply reverse engineering to get the details that might be needed for an

evaluation. Thus, satisfaction of the Common Criteria requirements on the composite system becomes increasingly difficult with higher assurance levels.

There are other technical issues in the development of reliable and secure software using existing components. A difficult issue is predicting the overall properties of the composite system, given information about its components. These properties include, but are not limited to, functionality, reliability, and security.

2. Standards

A number of standards for the evaluation of software and systems have been developed and tailored to different application domains and different concerns. Originally national in scope, many of these efforts have attained international support.

For secure systems, the preeminent standard is the Common Criteria, which were developed as a merger and reconciliation of several national standards. The avionics industry has its own standard, DO-178B [33]. Other industries use standards such as IEC 60601 (pertaining to medical electrical equipment) or IEEE 1012 (pertaining to software verification and validation). There is a U.S. standard, FIPS-140, for the evaluation of cryptographic systems. In this section we describe standards that we found pertinent to the project.

2.1 Common Criteria

The Common Criteria (CC) were developed in the 1990s as a unification of different national standards for the evaluation of software. Significant inputs were the U.S. Trusted Computer System Evaluation Criteria (TCSEC), Canadian Trusted Computer Product Evaluation Criteria (CTCPEC), and the European Information Technology Security Evaluation Criteria (ITSEC) standards. Version 1.0 of the CC was published in 1996. The most recent version, 2.2, was published in 2004. The CC version 2.1 has also been adopted by ISO as standard 15408.

In the CC, a *target of evaluation* (TOE) is assessed for meeting a *security target* (ST), which describes the security threats, objectives, requirements, security functions, and assurance measures relevant to the evaluation. These elements are usually drawn from a *protection profile* (PP), which defines a set of security requirements relevant to a type of application. Protection profiles have been developed for operating systems, firewalls, databases, and other types of system.

An evaluation results in an *observation report* and a *evaluation technical report* (ETR). Observation reports provide a mechanism for documenting problems encountered in an evaluation or to request clarifications. The ETR presents technical justification of the evaluation verdict (*i.e.*, whether the evaluation succeeds or fails). The evaluation might require access to confidential or proprietary information, and the ETR itself may contain such information. These results are then provided to a certification body, which reviews the output of the evaluation and, if appropriate, will issue a *certification report*.

The CC defines seven assurance levels, running from an Evaluation Assurance Level 1 (EAL 1) to Evaluation Assurance Level 7 (EAL 7), with increasing demands on the design, development, and analysis of the TOE. At EAL 1, the TOE is functionally tested and security threats are not viewed as serious. Levels 2 and above require the developer to provide design information and test results. Levels 5 and above require rigorous development practices and formal modeling.

A Common Evaluation Methodology (CEM) [2] is also defined, allowing developers, evaluators, and certifiers to agree on the evaluation process. The CEM applies to protection profiles, security targets, and systems. The CEM applies only to EAL levels 1 through 4. Some of the requirements of higher levels are not thoroughly understood, and agreement has not been reached on an evaluation methodology.

A community can establish an *evaluation scheme*, which sets out a framework for the application of the CC, and which typically establishes and monitors standards for the quality of evaluations. Many nations, including Canada, France, Germany, the United Kingdom, and the United States, have their own schemes.

The "Common Criteria Recognition Arrangement" describes how certifications by one country may be accepted by other member countries. However, the mutual recognition arrangements are for the lower levels of the CC hierarchy and do not go to EAL5 or above. In fact, there appears to be nationalistic concerns at the higher levels with countries maintaining proprietary approaches to these higher levels of security evaluation.

The CEM addresses composite systems in Section B.6, "TOE Boundary," which states that the TOE may be a part of a product, a complete product, a set of products, a unique technology, or a combination of these. This does not completely define the boundary, as some components (e.g., an LDAP server) might be considered as part of the environment for the TOE. However, the definition of "TOE security functions" implies that the TOE must include all components relevant to its security. Thus, if the LDAP server were to be used to store security information used by a TOE, it would have to be considered a part of the TOE. The rules for establishing a boundary are somewhat subjective and some evaluations have placed security relevant components outside the TOE.

The CC defines *components*, which are specific requirements. For example, component ADV_HLD.2 is a requirement for a "security-enforcing high-level design." These components are grouped into *families* such as ADV_HLD, which includes components ADV_HDL.1 through ADV_HLD.5 of increasing stringency. These families are themselves grouped into *classes*. The CC defines classes ACM (configuration management), ADO (delivery and operation), ADV (development) AGD (guidance documents) ALC (life cycle support) ATE (tests), and AVA (vulnerability assessment).

In order to avoid confusion between the CC meaning of the term "component" and its meaning as "an element of a compound system," we will use the phrase "CC component" or "assurance component" when the first meaning is intended.

Most of the CC classes are relevant to component-based software:

- Configuration management requirements (class ACM) are intended to ensure that only the intended elements are included in a system, and that all changes are properly authorized. These requirements might be deemed to apply to the components used in building a system; if so, the component developer's practices and tools might be under review.

- Delivery and operation requirements (class ADO) are meant to provide assurance that an evaluated system can be delivered, installed, and started in a way that conforms to the

assumptions of its evaluation and that maintains its security. Some of these installation and start-up procedures can apply to the components of a system, *e.g.*, in the case of a separate database server.

- Development requirements (class ADV) concern the implementation of the trusted security function (TSF). The TSF is represented at different levels of abstraction, and correspondence must be shown between the levels. The requirements are meant to provide assurance that the security requirements are properly implemented. For a composite system, there may be a need to document the high-level design (HLD), low-level design (LLD), and implementation (IMP) of any security-relevant components. Specifically:

  - HLD level 3 requires a level of detail for each subsystem: “The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing complete details of all effects, exceptions and error messages.” This complete detail may be hard to determine for a COTS component.

  - IMP level 2 requires the developer to provide the “implementation representation” for the entire TSF. This will include any security-relevant components.

  - Trusted Security Function Internals (INT) level 2 requires an architectural description that “shall describe how the portions of the TSF that enforce any access control and/or information flow control policies have been structured to minimize complexity.” This may require design information for security-relevant components.

  - LLD level 2 requires a low level design for each module, including “complete details of all effects, exceptions and error messages” for each interface. Like HLD level 3, this may require implementation details for security-relevant components.

  - Representation Correspondence (RCR) levels 1 and above require arguments of correspondence between the other descriptions (high-level design, low-level design, and implementation). Consequently, beginning at level 2, these arguments will be needed for security-related components.

- Life cycle support requirements (class ALC) concern the tools, techniques, and security of the development environment. Like the configuration management requirements, these requirements might be deemed to apply to the components used in the construction of the TOE, and if so, the component developer's tools, practices, and methods would be subject to assessment. Component developers might not want to reveal the required information. Of particular interest are the requirements on flaw remediation. Clearly, flaws that arise because of errors in a component might not always be corrected by modifying the TOE, and in such cases, the remediation procedures of the component developer are relevant.

- Testing requirements (class ATE) define four families of testing to which a TOE might be subjected. Some of these tests, such as `ATE_DPT.2`, require a detailed low-level design. To meet such a requirement with a component, the component vendor would need to supply such a low-level design, and either the vendor must also supply evidence of testing or the developer must test the component.

- Vulnerability assessment requirements (class AVA) describe several different sorts of vulnerability assessments. Several of these, such as AVA_CCA.2 (systematic covert channel analysis), explicitly depend on ADV_IMP.2, which states that "the developer shall provide the implementation representation for the entire TSF." Thus, if a component is part of the TSF, its source code will be required.

The requirements become more stringent at higher assurance levels. Table 1shows the correlation of "development" category (ADV) assurance measures with evaluation levels, using the names and numbers defined in the CC. Thus, for EAL4, measure ADV_FSP.2 is required. The shaded cells show the measures that may apply to components, as described above.

Table 1. Evaluation Assurance for Development (formal)

CATEGORY	EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
ADV_FSP	1	1	1	2	3	3	4
ADV_HLD		1	2	2	**3**	**4**	**5**
ADV_IMP				1	**2**	**3**	**3**
ADV_INT					1	**2**	**3**
ADV_LLD				1	1	**2**	**2**
ADV_RCR		1	1	1	**2**	**2**	**3**
ADV_SPM				1	3	3	3

Table 2 gives the same information using informal language. As can be seen from the table, at the lower assurance levels, detailed information for the components is not required, but at levels 5 and above, details of the components will be needed that go beyond what COTS components usually offer. Level 4 imposes the need to describe modules of the implementation, which may already cause problems for a composite system.

The CC differs from the other evaluation standards discussed here in its flexibility. The other standards apply to more specific application areas, and therefore impose specific requirements. In contrast, the CC are general, and in an evaluation the protection profile and security target define the specific requirements. This flexibility creates issues that the other standards do not share. For example, a component might be evaluated with respect to one protection profile, then be used in a system that is to be evaluated with respect to a different profile. Some argument then need to be made that the security threats considered for the evaluation of the component are appropriate for its use in the new system, which may be considering different threats.

Table 2: Evaluation Assurance for Development (informal)

CATEGORY	EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Functional specification	Informal			fully-defined	semi-formal		formal
High-level design		Descriptive	Security-enforcing		semi-formal(3)	semi-formal(4)	formal
Implementation of TSF				subset	full	structured	
TSF Internals					modularity	reduction of complexity	minimisation of complexity
Low-level design				descriptive		semi-formal	
Correspondence		Informal			semi-formal		formal
Security Policy Model				informal	formal		

2.2 DO-178B and ED-12B

Published by the Radio Technical Commission for Aeronautics (RTCA) in 1992, DO-178B is a standard for the certification of avionics software. The European standard, ED-12B, is the same document.

DO-178B describes specific requirements for avionics software, including allowable programming languages, features that may be used in these programming languages (*e.g.*, recursion is not allowed), development process, and the necessary analysis of source and object code. Analyses include detailed tracing of high-level requirements to low-level requirements, low-level requirements to source code, and source code to object code; a demonstration of complete code coverage in testing; and tests showing that all conditions in a Boolean test are needed.

DO-178B defines five different levels of criticality of software and systems, depending on the consequences of failure. The most critical systems, at level A, are those where failure prevents continued safe flight or landing; the least critical, at level E, have no effect on the aircraft's safety. Meeting “level A” requirements involves the use of specialized development tools and stringent testing.

2.3 FIPS 140

FIPS 140-2 is a U.S. Federal Information Processing Standard for the evaluation of cryptographic components, belonging to a suite of standards for information processing systems. The standard defines security requirements covering the design and implementation of a cryptographic module, and defines four levels of satisfaction of these requirements. FIPS 140-2 is the second version of this standard, replacing FIPS 140-1. Many of the requirements of the standard concern the hardware itself, including tamper resistance and tamper detection, and the quality of the "random" number generation. At levels 3 and 4, there are more stringent requirements on software documentation and review; level 4 requires formal mathematical modeling of the system and its security policy.

The IBM 4758 was validated to FIPS 140-1 level 4, the highest level, in 1998. The device is a programmable platform, and the developers hoped that other systems using it could take advantage of its validation [36]:

> "Since our device is a programmable platform, we hope this work substantially lowers the barrier for others to develop, deploy, and validate secure coprocessor applications."

Some other devices using the 4758 have been validated, such as the Pitney Bowes ClickStamp Online CCV (Cryptographic Co-processor for a Virtual Meter), as reported in FIPS certificate number 84. However, we have not been able to determine whether the hopes of the IBM developers were realized.

2.4 TCSEC, TNI, and TDI

The U.S. National Security Agency published a series of security evaluation standards in the 1980s, which have now been superseded by the Common Criteria. These standards offered mechanisms for evaluating composite systems, through two "interpretations" of the initial TCSEC criteria. The Trusted Network Interpretation described systems decomposed into peers, while the Trusted Database Interpretation applied to systems built in layers. Tinto [37] provides a concise overview of the standards and the framework they define for evaluations of composite systems.

2.4.1 Trusted Network Interpretation

The Trusted Network Interpretation [30] (TNI) is part of the United States' Rainbow series of standards for the security of information systems, published by the NSA in the 1980s. The bulk of the TNI is an elaboration of the Orange Book evaluation criteria, explaining how they apply to networked systems (as opposed to the monolithic systems originally considered). For the most part, the networked system is considered as a single evaluated system:

> "The interconnected accredited AIS (Automated Information System) view is an operational perspective that recognizes that parts of the network may be independently created, managed, and accredited. Where different accrediting jurisdictions are involved, the joint

> approval process is required, describing the handling practices and classification levels that will be exchanged between the components involved."

Interconnected accredited AIS consist of multiple systems (some of which may be trusted) that have been independently assigned operational sensitivity ranges (the highest and lowest sensitivity levels of information that may be simultaneously processed on that system). In this view, the individual AIS may be thought of as "devices" with which neighboring systems can send and receive information. Each AIS is accredited to handle sensitive information at a single level or over a range between a minimum and maximum level.

Section I.4.3, "Component Evaluation," describes a process of evaluation that reuses evaluated components, without requiring them to be re-evaluated:

> "Because network components are often supplied by different vendors and are designed to support standardized or common functions in a variety of networks, significant advantages can accrue from a procedure for evaluating individual components. The purpose of component evaluation is to aid both the network designer and the evaluator by performing the evaluation process once and reusing the results whenever that component is incorporated into a network."

Appendix A (of the TNI) discusses the case where a networked system's is viewed as the composition of trusted components. Among the issues considered is "How can the composition of rated components be evaluated?" Various "composition rules" for joining, (e.g., Discretionary Access Control (DAC) systems) are given.

Appendix C (of the TNI) describes conditions on the communications between components (specifically, any channel must have a single level, or must implement a labeling scheme so that data labels can be preserved across components), and identifies the "cascade problem." The cascade problem arises when weaknesses in different systems, each deemed to be of low risk, combine to create a high-risk weakness. For example, a set of systems, each operating at two adjacent security levels, might have a weakness allowing data to be downgraded. For a single system, this single level of downgrade might be relatively insignificant. In combination, however, the systems could allow multiple levels of downgrade to occur.

The appendix (of the TNI) offers some guidelines on determining whether a set of interconnected systems offers enough protections to overcome the risks of the cascade problem.

2.4.2 Trusted Database Management System Interpretation

The Trusted Database Management System Interpretation [29] (TDI) describes how the TCSEC evaluation criteria can be applied to hierarchical systems, where each layer depends on lower layers only. The prototypical application for such decomposition is a secure database system running on, and relying on the security of, an operating system.

> "Modern computing systems are rarely conceived and built by a single organization. Rather, the rule is that systems are constructed by assembling parts—hardware, firmware, and software—produced independently by various organizations or vendors. This fact introduces a fundamental difficulty into the task of evaluating a "system" for conformance to the trust requirements of the Trusted Computer System Evaluation Criteria (TCSEC). This difficulty stems from the fact that assessment (either evaluation of a product or certification of a system) entails a global perspective of the entire system under consideration. There are not yet widely accepted methods of factoring the various aspects of a trust assessment and then reassembling the results into a statement about the whole."

The TDI defines the notion of *TCB subsets*, a generalization of the Trusted Computing Base (TCB) concept. This generalization allows a TCB to rely of the services of a lower layer. The TDI describes how a system's policy can be decomposed into parts that can be implemented by the different TCB subsets.

2.4.3 TCSEC Evaluations of Composite Systems

Several systems were evaluated under the framework for composition. A list of evaluated products can be found at http://www.radium.ncsc.mil/tpep/epl/index.html. There are two notable examples of composite system evaluations:

- The Novell NetWare 4 Network System Architecture was successfully evaluated under the TNI, allowing components to be replaced or added without the need to evaluate the whole system. Specific Novell products in this architecture were evaluated, as was a third-party client system, SISTex Assure EC 4.11.

- Several database systems, including several versions of the Oracle Object-relational database, were evaluated under the TDI. Oracle was evaluated with respect to an underlying operating system that could be Solaris 8 or Windows NT 4. These systems had prior TCSEC evaluations, and did not need to be re-evaluated as part of Oracle's evaluation.

3. General Literature Survey

The literature survey was conducted by a combination of web searching using Google; bibliographic searches using bibliographic databases such as the ACM digital library, Citeseer (http://citeseer.ist.psu.edu), the Computer Science Bibliography Collection (at http://liinwww.ira.uka.de/bibliography/index.html); review of papers and proceedings of the Common Criteria conferences; and by following references from papers found by these sources.

3.1 Component-based systems

Research in component-based systems and component-based software engineering is ongoing. Recent work is reported at a series of ICSE workshops ([7,8,9]), a series of OOPSLA workshops ([15,16]), and in the DARPA CHATS project [31]. As is clear from this research, predicting the overall properties of a composite system remains a difficult problem. A simple example is in determining the reliability of a composition of two components, one acting on the other's outputs. The reliability of each component is measured in terms of a probability distribution on its inputs. Since the output of the first component is unlikely to have a distribution matching the random distribution of inputs assumed in the analysis of the second component, no overall reliability figure can be determined.

Research into the derivation of properties of composite systems from their components has a fairly long history. Security properties have been studied since the mid 1980s, with results by McCullough [27,28] and others showing how, for many definitions of “secure system,” connecting two secure systems results in an insecure system. The IEEE Symposium on Security and Privacy series of conferences shows over 20 years of progress in the area. Recent papers, such as [25], show that significant advances have been made, and that there are several reasonable security properties (e.g., non-interference and separability) that are preserved, or even emerge from, careful composition.

Component-based software is widespread. Well-developed frameworks, such as CORBA, Java Beans, COM, and DCOM, provide mechanisms for combining software components into a system, and there is an industry devoted to “middleware” software that can be used to integrate distributed components. There is a significant gap, however, between the tools used in practice and the emerging research results.

3.2 Common Criteria

The CC recognizes the desirability of re-use of evaluations. A 2002 Information Notice [4] discusses a scenario in which “an evaluation is considering utilizing a previously completed evaluation in the evaluation of another compound TOE (i.e. a TOE in which the previously evaluated TOE is incorporated).” A specific issue acknowledged in this notice is that the evaluation information may be proprietary, and that special contractual arrangements may be required so that this information can be acquired.

Re-usability is one of the “universal principles of evaluation” discussed in CEM-97/017 [3], an unfinished draft that was apparently abandoned in 1997. This report reveals some of the thinking of the developers of the CC.

While the virtues of re-use of evaluation are recognized, specific mechanisms for this re-use are not well-defined, and experience in applications has exposed problems. Galitzer [14] reports on the difficulty of reconciling and rationalizing multiple PPs and difficulties with the sheer number of documents required for a composite system.

A *precedent database* maintained by NIST under the U.S. CC scheme records some decisions relating to the evaluation of composite systems. These are discussed in Section 6.2.

The Joint Interpretation Working Group is composed of IT certification experts from France, Germany, the Netherlands and the United Kingdom. This group has defined an “ETR-lite” [19] intended specifically for use in evaluation composite systems. The idea is for complete, detailed documents to be used in an evaluation, and for a set of reduced documents to be made available with the component (under “commercial-in-confidence” terms). These “lite” documents do not include sensitive proprietary information, but are meant to include enough information for users of the component to complete an evaluation of their systems.

The UK has established a working group to study composition [6]. The group has focused on a subset of the problem, listing remaining issues for later work in an appendix. As this group is at an early stage, the work identifies issues and problems rather than solutions. Among the issues raised are

- The evaluated version of a component might not include some functionality that is used in the composition.
- A component in a system might be privileged with respect to the policies enforced by other components.
- A component's TSF interfaces, as defined for its evaluation, might not include the programmatic interface.
- What mechanisms can be used to pass information from the component's evaluators to the composite's evaluators?
- The evaluated configuration of a component might not match the requirements of a composite.
- Who bears the extra evaluation costs, the component developer or the composite developer?
- Is additional testing of a component required when it is used in a composition?
- Who is responsible for configuration management of the components?

- What happens if a new vulnerability is discovered in a component? (The CC defines assurance continuity and flaw remediation mechanisms, but these are not directly applicable to compositions).

- How can the requirements of a PP be distributed over a set of components?

The group intends to propose changes to the CC and CEM, and new constructs for use in a PP, to make them more suitable for composite systems.

Perhaps the greatest success in composite evaluation under the CC has been attained in the Smart Card industry. Smart Cards are designed and manufactured by collaboration between the hardware and software developers, who are usually different organizations. Neither organization has complete information about the other's product, and typically neither wishes to share their proprietary information with the other. von Faber [10] describes a methodology (using ETR-lite) that has been used to protect both parties while satisfying the evaluators. Ichihara [18] describes the construction of a composite ST derived from the ST-lite of hardware and the PP of the software. This is done by making the input documents sufficiently flexible that many interpretations are possible, with the merged document resolving some or all of this flexibility. However, even though these projects are described as successes, the reports identify numerous remaining difficulties in carrying out evaluations of composite systems.

Karger and Kurth [21] argue that the ETR-lite approach is inadequate for higher levels (EAL5 and up), because too much information is deleted. They recommend that the full ETR and the developer documentation for the hardware component must be made available (under non-disclosure agreement) to the software developers. Their paper relates many examples of issues in hardware and firmware that have security relevance for software, yet would not appear in the ETR-lite. Karger and Kurth also note that the very notion of composite evaluation is controversial and is believed by many to be impossible. Clearly, these opinions are at odds with ongoing work in the Smartcard community.

3.3 DO-178B

Issues surrounding the use of COTS software in avionics systems were surveyed and reported to the United States Federal Aviation Authority in 2001. The report [22] also discusses how COTS is handled in other regulated application domains, such as elevator control, medical devices, nuclear control, and space. All these industries, and all the regulators, are struggling with the problem; all are sensitive to the pressures to use COTS components, and none has a satisfactory way of dealing with COTS. The report concludes that COTS components might be applicable to the lower levels (C and D) of criticality, but are unsuited for the higher levels. Surveys with application developers in all five of the domains indicate strong reluctance to use COTS. The medical industry has the highest acceptance of COTS components; this may be because of the relatively high volumes and high profits in that sector, which motivate COTS vendors to provide evaluation evidence along with their software.

The DO-178B standard has been in use for many years, and a number of clarifications have been required. As Romanski [32] discusses, many of these concern the use of “previously

developed software" (including COTS). A mechanism has been put in place to allow "certification credits" to be granted for components that have been used in systems that have already been evaluated. However, many issues remain [5]:

- Components cannot be certified in isolation, only as part of a complete system.
- A component may be evaluated to a lower level than is required of the new system that uses it.
- A component vendor might not be willing to provide the documentation needed for certification of a system using the component.

As Budden [5] discusses, it is often necessary to collaborate with the COTS vendor in order to certify a composite system. This may require special agreements with the vendor, if proprietary information must be shared.

Even though the certification process does not yet fully support the development of certified components or the certification of composite systems without re-evaluation of its components, the industry has managed to develop mechanisms that enable the use of components, with the vendors assuming the burden of providing certification evidence. One company, Validated Software (validatedsoftware.com) sells DO-178B validation components. These are software components with full source code, test scripts, and documentation as required for the customer to include the component in a system to be certified. Similarly, Green Hill Software (www.ghs.com) offers components such as an operating system and an Ada run-time system, as kits with all the materials needed for certification. This is very much a "white box" approach to reuse. Other companies sell specialized tools for use in developing avionics software; for example, Aonix (www.aonix.com) sells a combination of compiler and run-time system suitable for DO-178B level A.

4. General Tools and Techniques Overview

4.1 ETR-lite Approach

The ETR-lite approach[19,10] was developed during some smartcard evaluations, as a way of dealing with the various components involved. A typical smartcard combines hardware, an operating system, and application software, which may be developed by different parties who are unwilling to share full details of their products with one another.

The Joint Interpretation Working Group has accepted the approach, and documents defining it are under consideration for the Common Criteria. The key idea of the approach is to define a set of less detailed documentation derived from the normal CC evaluation documents. The ETR-lite omits proprietary information or information relevant to national security from the ETR; similarly, the ST-lite omits detail from the ST.

Several composite evaluations have been completed using the ETR-lite approach, including the Gemplus GemCB-B0'/EMV, using hardware (a P8WE6004) developed and evaluated by Philips.

4.2 Composite-ST Approach

Ichihara [18] describes the evaluation of the Japanese JUKI smartcards. The card's hardware was evaluated in France with respect to a French PP (PP9806). A specific profile, JUKI-PP, existed for the complete system, and an ST-lite for the hardware was available.

The challenge in this evaluation was to develop a ST for the smartcard that conformed to the JUKI-PP. Ichihara describes how the ST-lite and JUKI-PP are combined and then modified to form the composite ST, and proposes a methodology for such combinations.

4.3 Engineered Composition Approach

Galitzer was funded by a Critical Infrastructure Grants Program (CIPGP) grant from the U.S. National Institute of Standards and Technology (NIST) Computer Security Division, to validate the *engineered composition* concept and to work towards its inclusion in the CC. As summarized in [14],

> "Engineered Composition is a top-down, horizontal decomposition approach, based on information systems object-oriented modeling principles and techniques for large systems that require distributed development. In our experience, it improves the way requirements and capabilities for systems and their components are expressed using the CC because it provides a means to write PPs and STs for component parts that are combinable and comparable. This has a number of advantages:

- Making PPs (and STs) combinable provides a means to capture the system properties of the composed components.

- Making PPs (and STs) comparable enables them to be composed without first being rationalized.

- Through reuse, realizing efficiencies for evaluating the security of systems, in terms of the number of PPs required and the effort to produce them and their respective STs."

The engineered composition approach treats the composition problem as a modeling problem with security as its subject domain. Therefore, it promotes the use of various modeling techniques to deal with the composition problem. Most significantly, object-oriented frameworks, design patterns, and aspect-oriented programming approaches are recommended. Galitzer has experimented with the approach by defining a framework for a public key infrastructure with a directory component. Defining the framework is reported to be relatively straightforward, while representing the framework in a format suitable for CC evaluation is difficult. Galitzer therefore recommends defining a CC composition framework to facilitate the expression of such designs. The framework includes an ability to use distinct *views* of a system to discuss different areas of interest, such as threats, policies, and components.

Engineered composition is at present just an idea, and aside from Galitzer's experiments with a PKI system, has not been extensively tested.

4.4 Evaluation Kits

An *evaluation kit* is a COTS component supplemented with whatever materials are required by a regulator. Evaluation kits are available for several components for DO-178B validation and for U.S. Food and Drug Administration (FDA) accreditation.

An evaluation kit differs from a normal component, in that additional documentation (such as source code, test results, and design documentation) and evidence is provided. In addition, the vendor may provide assistance in ensuring that the regulator accepts the evidence.

Evaluation kits are available in cases where the regulatory requirements are well defined. For the CC, there is a problem because different STs may require different evidence. Thus, it is unlikely that a simple CC evaluation kit can be defined; instead, a vendor would need to provide a kit for a specific PP or ST. This may limit the size of the market for such kits, making them expensive.

4.5 Regulatory Data Banks

The medical devices industry and regulator (FDA) in the United States have a noteworthy relationship. The regulator maintains a repository of detailed information about COTS products, using data supplied by the vendors. This data is kept confidential, so vendors need not be concerned about revealing proprietary information to their competitors. The FDA can

use the data when assessing medical devices that use these COTS components. We have been unable to find specific details as to this approach.

The medical industry may not be a good model for CC validations. As discussed in [22], medical devices have high volumes (e.g., Krodel reports devices with tens of thousands of units) and markups are high. Vendors are therefore willing to supply validation evidence in order to sell into this market.

A regulatory data repository with evaluation evidence supplied by COTS vendors would differ substantially from an evaluation kit. An evaluation kit puts all the evaluation evidence in the hands of the purchaser, who adapts it as needed and furnishes it to the evaluator. By contrast, evidence in a regulator's repository is not available to the user of a COTS component, and cannot be adapted or modified by them.

An indication that the use of regulatory data banks may be a trend is exemplified by a NIST data bank, called the national software reference library, containing electronic voting software. This suggests that for several markets, companies are willing to lodge proprietary technologies with government regulators.

5. Current Practice

For the information that follows, Mr. Craigen made use of interviews and email. Unfortunately, not as much information as was hoped for was forthcoming.

5.1 CSE

Mr. Craigen visited with CSE and through interviews concluded that CSE has no current activities relating to composite evaluation.

5.2 USA

The United States CC evaluation and validation scheme is jointly managed by the NIST and the NSA. Among the information maintained is a "precedent database" (at http://niap.nist.gov/cc-scheme/PD/index.html) and an interpretations database (at http://niap.nist.gov/cc-scheme/PUBLIC/index.html). These databases contain several items pertaining to composite systems, such as PD-0004 (Satisfaction of Requirements by Applications Running on Untrusted Products), PD-0046 (Exclusion or Inclusion of an Operating System in the TOE), PD-0050 (How Should Libraries Be Handled Relative to the ADV_FSP.1 work units of the CEM), and PD-0101 (Level of Detail Necessary for Assurance Requirements on Third Party Products). These interpretations discuss how the operating system, libraries, and third party components, are to be treated. In all cases, the answer is the same: the entire product must be evaluated, and any security-relevant components need to be considered regardless of their origin.

Precedent PD-0100 (See Section 6.1.2), "When can evaluation evidence be reused?" discusses impediments to the reuse of evidence, particularly when new CC interpretations apply to the old evaluation, or when there is any security-relevant change to the TOE.

Mr. Craigen identified that there is a NIST *Ad Hoc* Working group on the Composability of Evaluated Products. However, inquiries with contacts at NIST did not lead to any useful information about the activities of the working group.

5.2.1 NSA

Regarding current activity at NSA, no information was acquired from Mr. Craigen's contacts at the Agency.

5.3 UK

As described above in Section 3.2, the UK is leading a technical working group on composition. This group has identified a number of issues in evaluating composite systems, and intends to propose changes to the CC and CEM in support of such evaluations. The group

also intends to propose new constructs for use in a PP to make them more suitable for composite systems.

5.4 Rest of Europe

The Joint Interpretation Library, discussed in Section 3.2, is composed of several European countries (including the U.K., Germany and France), and has published standards relating to the ETR-lite for composition. The Europeans are particularly involved in the evaluation of Smart Cards (see Section 6.1.1).

The European COTS User Working Group (ECUA) has published a report "Software Reuse in Safety-Critical Applications" (http://www.esi.es/en/Projects/ecua/pdf/Final-Summary-Report.doc).

5.5 CBIS

Mr. Craigen contacted members of the government side of the CBIS evaluation team. During one of Mr. Craigen's visits to Washington a meeting was scheduled to discuss the CBIS evaluation. Unfortunately, the CBIS folks were called away to another meeting and so the interviews did not happen. Follow up efforts have been fruitless in part, because the prime contractor has now taken the view that their approach to composition is proprietary.

5.6 NATO

NATO has had working groups considering areas of relevance to composite systems and their evaluation. IST-018/RTG-005 studied the use of COTS in military systems; information is available at http://seg.iit.nrc.ca/nato/ .

IST-027/RTG-009 was the NATO Research Task Group on Validation, Verification and Certification of Embedded Systems, chaired by Dan Craigen. Certification was considered, but composability was not specifically addressed. Further information is available from http://www.ora.on.ca/nato/.

Reports for both NATO efforts should ultimately be lodged on the NATO website: www.rta.nato.int.

Mr. Craigen also approached the Chair of the IST Panel and Morven Gentleman (now at Dalhousie University) and, from these sources, concluded that the above activities were the most material to the subject matter of this project.

6. Conclusions

In this concluding section, we bring together our observations regarding evaluation of composable systems. Reflecting the terms of the project, this section is divided thusly:

1. Assessment of Current Practice
2. Summary of Current R&D Trends
3. Technique Omissions, Challenges, Gaps
4. Adequacy of Current Evaluation Techniques
5. Evaluating the Secure Access Manager

6.1 Assessment of Current Practice

Our general opinion is that while there have been successes in evaluating composite systems, there are numerous issues and trouble spots, some of which are discussed below. Evaluating composite systems, even at the lower levels of the Common Criteria, appears to be costly, performed by large, well-funded organizations (e.g., Microsoft, IBM, Oracle) and requiring heroic effort. Even though some complex systems have been evaluated up to EAL4+ (e.g. Windows 2000 Professional, Server and Advanced Server with SP3 with Q326886), there is a sense that the approaches do not scale in a satisfactory manner. If they did scale or were cost effective the list of evaluated products would be of much greater magnitude.

6.1.1 Nested Protection Profiles

One of the problems evaluating composite systems, even when the components have been successfully evaluated under the Common Criteria is that the components are normally evaluated against different Protection Profiles. The protection profiles identify the security environment, security objectives, security functional requirements and security assurance requirements. These profiles tend to be quite different in scope. However, one approach used in the context of Smart Cards is to use “Nested Protection Profiles.” This approach seems to work within the Smart Card domain since Smart Cards tend to follow generic architecture and the various components are really only targeted for this particular application domain (unlike, for example, Microsoft Windows or the Oracle database).

For Smart Cards, three Protection Profiles were defined:

- PP9806: Smart Card Integrated Circuit Protection Profile
- PP9911: Smart Card Integrated Circuit with Embedded Software
- PP0010: Smart Card IC with Multi-Application Secure Platform

In the top level Protection Profile (PP0010) the nesting is shown through a number of quotes. For example:

> "A Security Target compliant with this PP shall claim the compliance to the PP9806. Indeed, this PP shall not be used independently."
>
> "This Protection Profile adds … to the PP9806:
>
> - The requirements of the Operating System software embedded in the Smart Card Integrated Circuit, (same as PP9911)
>
> - Necessary requirements to assure the security of the Smart Card IC with Multi-Application Platform and mostly of the Loaded-Application system interface"
>
> "Evolution of Smart Cards toward Multi-Application platform as well a multi-layer architecture leads to additional requirements. This PP is upwardly compatible with the PP9806 and PP9911 but provides extensions…"

6.1.2 PD-0100

In Section 5.2, we discussed briefly U.S. interpretations that are lodged in their precedence database. For this project, it is of interest to note that a few of these precedences are generally negative as they pertain to evaluating composite systems. Just because another vendor develops a component of a systems does not mean that the current developer is free from having the component evaluated. As concisely described in one document "The TOE is the TOE." The boundary of the TOE must include all the security relevant functions.

6.1.3 Repositories

As discussed in Section 4.5, the U.S. FDA has set up a regulatory databank consisting of detailed information about COTS products, using data supplied by the vendors. This data is kept confidential, so vendors need not be concerned about revealing proprietary information to their competitors. In a similar vein, according to an online article published by Wired (see www.wired.com/news/evote/0,2645,65490,00.html), five voting machine makers have agreed to submit their software programs to the U.S. National Software Reference Library for safekeeping. For the 2004 U.S. presidential election, the stored software can be used as a comparison tool for election officials. The article goes on to state that the National Software Reference Library is part of an election security initiative that was launched by the U.S. Election Assistance Commission. The deposited software will cover 90% of the software used in computerized voting machines. These two cases (FDA and EAC) suggest that the use of government repositories may be a means for sharing proprietary information for component evaluations. However, we suspect that this approach may work more at a national level than at an international level.

6.1.4 Inclusiveness, not Simplification

One final remark regarding current practice and, in particular, standards. It is our opinion that standards do not strive for simplification, but generally strive for inclusiveness. The price for inclusiveness is complexity. For example, systems tend to support numerous overlapping mechanisms and oftentimes all these mechanisms must be included in associated standards. From a networking perspective we have protocols based on TCP/IP (Internet Protocol) and Novell. From a cryptographic perspective, one has such mechanisms as Secure Socket Layer (SSL), Secure Shell (SSH) and IP Security Protocol (IPSec). The interaction of system complexity and standards complexity makes it significantly more difficult to determine the applicability of evaluation techniques and for properly structuring systems to ease evaluation.

6.2 Summary of Current R&D Trends

We did not find a substantial body of work directed towards the evaluation of composite systems. One source that was useful, however, was the ICSE Workshops on Component Based Software Engineering (years 2001 through 2003). Indications of pertinent work are suggested by the following paper titles:

- Experiences with the Certification of Reusable Components in the GSM Project in Ericsson
- Trusted components: Towards Automated Assembly with Predictable Properties
- Compositional Performance Reasoning
- Correct and Automatic Assembly of COTS Components: An Architectural Approach
- Issues of Predicting Reliability of Composed Components
- Component Verification and Certification in NASA Missions
- Ensuring General Purpose and Domain Specific Properties Using Architectural Style

In our opinion, there is a significant gap between the tools, techniques and requirements used in practice and emerging results. In fact, our opinion is that most research work focuses on the Common Criteria component ADV_RCR; the massive issues pertaining to documentation and traceability are hardly touched.

6.3 Technique omissions, challenges, gaps

As noted in the introduction to Chapter 6, our general opinion is that while there have been successes in evaluating composable systems, there are numerous issues and trouble spots.

Evaluating composable systems, even at the lower levels of the Common Criteria, appears to be costly, performed by large, well-funded organizations (e.g., Microsoft, IBM, Oracle) and requiring heroic effort. Even though some complex systems have been evaluated up to EAL4+

(e.g. Windows 2000 Professional, Server and Advanced Server with SP3 with Q326886), there is a sense that the approaches do not scale in a satisfactory manner. If they did scale or were cost effective the list of evaluated products would be of much greater magnitude.

6.3.1 Protection Profiles

A particular area of concern is that of Protection Profiles. We do not have the technology for either generating Protection Profiles for system components from the Protection Profile for a composite system; nor, has there been much success in composing Protection Profiles. Basically, reconciling Protection Profiles is a difficult task. In its most general terms an algebra for Protection Profiles would be helpful, but not likely addressable any time soon.

6.3.2 Plugins

Many systems make use of plugins, yet the evaluation of plugins seems, at best, to be in a nascent state. We did not find any approaches on determining what the Protection Profile for a plugin would be since, in general, one does not know the threat environment, security objectives, etc., of the locality of the plugin. However, it may be possible to make use of the nested Protection Profile approach (discussed in Section 6.1.1) to help in the evaluation of plugins.

6.3.3 Assurance Continuity

To make evaluation more cost effective and broadly applicable, techniques need to be in place to allow for incremental changes to evaluation (assurance continuity) resulting from incremental changes to software.

The Common Criteria has been extended to take into account assurance continuity. So, for example, we have from "Assurance Continuity: CCRA Requirements," Version 1.0, February 2004, CCIMB-2004-02-009:

> "When a change to a certified TOE has been determined to be of major impact, the implication is that a more concerted analysis, and by an independent evaluator, is required to assess the assurance of the changed TOE. A re-evaluation is performed in the context of an earlier evaluation, reusing any results from that earlier evaluation that still apply."
>
> "It is possible that the developer may opt directly for re-evaluation without ever creating an Impact Analysis Report (for example, if the changes are so substantial that the changed TOE bears only a minimal resemblance to the evaluated TOE). Alternatively, even with substantial changes, the developer still may have conducted a security impact analysis of the differences between the changed TOE and the evaluated TOE."

As noted, the Common Criteria has recently had some changes allowing for incremental evaluation, the approach seems to be immature; in effect, assurance continuity is an emerging capability.

6.3.4 Commercial Practice

It is readily apparent that general commercial practice in system development is at odds with evaluation practices (at least, to some extent). The pressures of the market place, the need to make rapid changes, the imperative of time to market, the need for broad functionality over assurance all trump technical concerns for assurance. Yet, in our opinion, there is value to be derived, even in a purely commercial world, from some aspects of the evaluation process. Protection Profiles, in particular, are excellent security targets (if done right) for identifying the security environment, security objectives, security functional requirements and security assurance requirements.

6.3.5 Architectures

Applicable composable architectures are not well understood. From looking at the Secure Access Manager (further discussed below) and current evaluation practices, the choice of architecture is important. For example, the use of an access mediator to the Oracle database is likely an approach that significantly simplifies the TOE. However, tweaking Samba (netbios SMB server) or using privileged interfaces will not be as amenable to evaluation.

6.4 Adequacy of Current Evaluation Techniques

The (in)adequacy of current evaluation techniques are generally shown above. In particular, we noted above that:

> Our general opinion is that while there have been successes in evaluating composable systems, there are numerous issues and trouble spots, some of which are discussed below. Evaluating composable systems, even at the lower levels of the Common Criteria, appears to be costly, performed by large, well-funded organizations (e.g., Microsoft, IBM, Oracle) and requiring heroic effort. Even though some complex systems have been evaluated up to EAL4+ (e.g. Windows 2000 Professional, Server and Advanced Server with SP3 with Q326886), there is a sense that the approaches do not scale in a satisfactory manner. If they did scale or were cost effective the list of evaluated products would be of much greater magnitude.
>
> Applicable composable architectures are not well understood. From looking at the Secure Access Manager (further discussed below) and current evaluation practices, the choice of architecture is important. For example, the use of an access mediator to the Oracle database is likely an approach that significantly simplifies the TOE; However,

tweaking Samba (netbios SMB server) or using privileged interfaces will not be as amenable to evaluation.

In some instances, the benefits of an innovation will outweigh the assurance impediments. Coming from a formal methods background one analogy comes to mind. In the early 1980s, researchers at the University of Texas at Austin developed the Gypsy Verification Environment. For its time, this was a superb piece of R&D demonstrating that code verification proofs were possible (though at some cost). In engineering the system they downplayed the assurance arguments pertaining to the mathematical logic they used. From an R&D perspective, this was fine. However, once the GVE started being used by the U.S. government on security critical applications, it was not enough to live with the R&D objectives, evaluation of the logic was necessary. As Craigen and Saaltink demonstrated in the 1980s, there were a number of logic and language pathologies with the GVE, which could result in the assertion FALSE being proven: not a good thing! However, careful use of the GVE could, in large measure, circumvent the difficulties. So, there is an ongoing tension between functionality and assurance. We also see this in a much broader scale with the U.S. Navy work on Aegis. Aegis is a substantial system, but the U.S. Navy performed a risk analysis with regard to threats and are adopting COTS products to obtain the requisite functionality.

6.5 Evaluating the Secure Access Manager

Despite the difficulties in performing evaluations of composite systems, there is hope that the Secure Access Manager (SAM) could be evaluated to EAL2 or EAL3. These levels avoid the need to delve into the implementation details of third-party components.

In our opinion, some important benefits could accrue from evaluation to levels EAL2 or EAL3. For example, IBM's Tivoli Access Manager was evaluated to EAL3. Blakley and Kurth report [46] several benefits of the evaluation, including:

- Thorough analysis of the security function, including the definition of the configuration providing high security.
- Identification and repair of several security flaws as a result of the analysis.
- Improvements to the administrator guidance, including the removal of inconsistencies and the addition of missing details.
- Improvements to the description of the installation procedure.
- Clearer descriptions and guidance of the security function of the product.

It is apparent, from our survey, that the evaluation of complex systems is possible, though not necessarily probable.[1] There have been evaluations, for example, of Windows 2000 and of

[1] A full list of endorsed CC products can be found at http://www.commoncriteriaportal.org/public/consumer/index.php?menu=4.

various Linux distributions. However, these evaluations were carried out by large organizations and the cost and time spent is not known and, likely, problematic.

There are several issues that must be considered in an evaluation of SAM. We discuss some of these issues in the sequel.

6.5.1 TOE Boundary

What is evaluated? Which components are part of the TOE, and which are in the environment? This could have a major impact on the level of effort. If the SAM is to be evaluated, the definition of the TOE will be a critical factor in limiting the scope of effort.

The CC offers some flexibility in defining the TOE boundary, even though it seems to say that all security-relevant functions must be inside the boundary. For example, the Tivoli Access Manager was evaluated to EAL3, with a TOE that does not include the LDAP directory storing information about users and groups. This component has obvious relevance to the security of the system, as alteration of the groups containing a user could lead to granting the user unintended access. There is an environmental assumption that the directory is protected from unauthorized modification.

Some components of the SAM might be dealt with in a similar way, especially in cases where there are no modifications to the component. The Active Directory and Mail servers might be examples of such components.

Clearly, the way in which the TOE boundary can be defined depends critically on the architecture of the system. An architecture that keeps the security mechanisms relatively small and avoids relying on the security functions of its COTS components should allow those components to be outside the TOE boundary. For example, the SAM POC documentation suggests that all accesses to the Oracle database use the custom client, which sends its requests to the Oracle custom PEP. The PEP checks the access control rules, then forwards the request to the Oracle database only if appropriate. This should allow the Oracle database itself to be outside the boundary, as its security functions appear to play no role in meeting the SAM's security objectives. This type of architecture, using security-enforcing wrappers, should significantly reduce the size of the TOE. In contrast, the custom PEP for the Samba server is integrated into the server itself, making it likely that the entire Samba server will be in the TOE.

The TOE boundary will also depend on exactly how the security objectives of the SAM are characterized. If, for example, objectives include controlling access based on labeling of data, then the functions of the Rights Management System and Oracle Label Security will be relevant to these objectives, and these components will be inside the TOE boundary. Alternatively, the objectives might be characterized as managing the configuration of these other systems. While this is objective is farther away from user-level requirements than the first, it might make evaluation easier because the TOE will not need to include the Rights Management System and Oracle database.

6.5.2 Protection Profile

It is not necessary to use a protection profile in an evaluation, However, if a protection profile is applicable, it would likely facilitate the definition of the security target for the evaluation, by defining many of the terms, concepts, threats, and mechanisms to be used. For example, the EAL2 evaluation of SuSE Linux did not claim performance to a profile, but it relied on parts of the Controlled Access Protection Profile.

Several protection profiles exist that might be relevant to the SAM. All these profiles can be found in the common criteria portal:

- Controlled Access Protection Profile. This profile is derived from the TCSEC C2 level requirements. The profile assumes a non-hostile, well-managed environment, with discretionary access controls for users accessing information. There are also requirements for identification and audit mechanisms.

- Labeled Security Protection Profile. This profile is derived from the TCSEC B1 level requirements. The profile is similar to the Controlled Access Protection Profile, but adds some mandatory access controls based on subject and object classification labels.

- Discretionary Information Flow Control (MU). This profile defines an environment of users and administrators who are indifferent to security, and a TOE administrator who is careful. The model applies to flows of data to and from depositories, e.g., emails from a mail server.

- Role-Based Access Control Protection Profile. This profile defines a set of requirements for a system offering role-based access controls. It includes a requirement to support separation-of-duty, which may not be appropriate for the SAM. It also requires a hierarchical set of roles.

In addition, there are several profiles for Oracle databases, and several for smart cards and readers.

6.5.3 Security Target

A security target is required for an evaluation, whether or not a protection profile is used. The CC specifies that a ST must have the following structure:

- TOE description

- TOE security environment: assumptions, threats, organizational security policies

- Security objectives for the TOE and for the environment

- IT security requirements

- TOE summary specification (security functions, assurance measures)

- PP claims (PP reference, tailoring, additions)

- Rationale (evidence to support the claim that the ST is complete and cohesive, that the TOE provides an effective set of measures in the specified environment, and the summary specification addresses the requirements)

This document is useful beyond the evaluation. In writing it, the developers need to document many of their assumptions about the system and its environment, and in the rationale can explain their reasons why the design of the TOE is adequate.

The ST is itself subject to evaluation. Among the requirements to be checked by the evaluators, the most significant are:

- ASE_OBJ.1, which requires the developer to trace security objectives to the threats and assumptions;

- ASE_REQ.1, which requires the developer to show that the IT security requirements meet the security objectives, and

- ASE_TSS.1, which requires the developer to show that the TOE summary specification and Functional Requirements agree (i.e., to show that the IT security functions meet the TOE security functional requirements).

As noted above, the security target's description of the security objectives has an effect on how the boundary of the TOE can be defined. There is some tension, therefore, between the goals of having a security target that is framed in terms of the user's needs, and one that describes the system in a way that will make the evaluation easier.

6.5.4 Evaluation evidence

The CC defines various “assurance components” for different assurance levels. We summarize here the components that apply to EAL2 and EAL3.

6.5.4.1 EAL2

The assurance components for EAL2 are shown in Table 3. Most of these elements should be in place for a professionally developed product.

6.5.4.2 EAL3

In addition to the assurance components for EAL2, several new assurance components apply to level 3, as shown in Table 4.

This level requires more effort on the part of the developers to ensure that the build environment is secure, and requires greater attention to the security functions of the TOE.

Table 3: EAL2 Components

AUT_CAP.2	Use of CM system, version numbers, and configuration items
ADO_DEL.1	Documented delivery procedures
ADO_IGS.1	Documented installation, generation, and start-up procedures
ADV_FSP.1	Informal functional specification of the TSF and its external interfaces
ADV_HLD.1	Informal high-level design of the TSF in terms of subsystems
ADV_RCR.1	Demonstrate correspondence of the HLD and FSP
AGD_ADM.1	Administrator guidance documents: usage and secure usage
AGD_USR.1	User guidance documents: usage and security notes
ATE_COV.1	Test the TOE against its functional specification
ATE_FUN.1	Test plans, procedures, expected results, actual results, for security functions
ATE_IND.2	Independent testing, including independent execution of developer tests
AVA_SOF.1	Show "strength of function" claims in the ST are satisfied
AVA_VLA.1	Vulnerability analysis

6.5.4.3 EAL4

EAL4 adds some components that are likely to be difficult for the SAM. Most significant among these is ADV_LLD.1 (descriptive low-level design, in terms of modules). These "modules" are meant to be at a much finer level of detail that “subsystems,” and meeting this requirement would entail describing the structure of any of the third-party components that are in the TOE.

Table 4: EAL3 Components

ACM_CAP.3	CM controls to ensure only authorized modifications, additional documentation
ACM_SCP.1	The TOE “implementation representation” and evaluation evidence must be under CM
ADV_HLD.2	Describe the separation of the TOE into TSP-enforcing and other subsystems, and identify externally-visible interfaces
ALC_DVS.1	Development security: ensure the integrity of the TOE design and implementation
ATE_COV.2	Systematic testing of the TOE against its functional specification
ATE_DPT.1	High-level testing of subsystems
AVA_MSU.1	Examination of guidance (this is evaluator effort, the developer has already produced these documents in AGD_ADM.1 and AGD_USR.1)

6.5.5 Use of Existing Evidence and Techniques

Some of the systems used in the Secure Access Manager Proof of Concept (SAMPOC) demonstrator have been evaluated. However, these previously evaluated systems are likely not being used in their evaluated configurations, or the evaluated security mechanisms are not being used. For example, Windows 2000 has mechanisms for discretionary access control based on Access Control Lists (ACL) or file encryption. Installation of components such as antivirus software or the windows rights management client might violate the administrator guidance (which may include an admonition not to install software with special privileges). Antivirus software scans all incoming mail, for example, and must be trusted to not divulge the contents. Similarly, the Keyboard Video Mouse (KVM) software and hardware used in the SAMPOC II mediates interactions between an administrator and several of the systems; therefore it can tamper with the “secure path” between them.

The ETR-lite approach might work for the EAL2 or EAL3 levels we have recommended. However, none of the component systems has produced these “-lite” documents.

The existing research results on composite systems and composition of security properties would have relevance to the ADV_RCR assurance component, where it must be shown that the high-level design of the TOE meets the requirements. Our review of the SAM POC architecture suggests that these results are not likely to apply to the SAM. Many of these results are meant to show how individually “secure” systems can be assembled to give a

"secure" result. In the SAM, however, the security of the overall system relies on the way in which the parts work together.

6.5.6 Summary

While there are some potential difficulties a SAM-like system could likely be evaluated within the Common Criteria to an EAL3 level. If such is to be considered for a deployable version of the SAM, we recommend that the developers interact with one of the Evaluation Labs as early as possible.

References

1. Common Criteria for Information Technology Security Evaluation. Version 2.2, January 2004. Available from www.commoncriteriaportal.org

2. Common Evaluation Methodology, version 2.2. January 2004

3. Common Evaluation Methodology for Information Technology Security, Part 1: Introduction and general model (draft) CEM-97/017

4. *Reuse of Evaluation Results and Evidence.* Common Criteria Information Notice 2002-08-009, unofficial, Oct 2002. www.commoncriteriaportal.org/public/developer/index.php?menu=5

5. Timothy J. Budden, AVISTA, Incorporated. *Decision Point: Will Using a COTS Component Help or Hinder Your DO-178B Certification Effort?* In STSC CrossTalk, Nov 2003. http://www.stsc.hill.af.mil/crosstalk/2003/11/0311budden.html

6. Common Criteria technical working group. *CC Project Technical Work Group: Composition* Draft version 0.2, June 2004

7. Ivica Crnkovic, Heinz Schmidt, Judith Stafford, and Kurt Wallnau. (eds) *Proceedings of the 4th ICSE Workshop on Component-Based Software Engineering: Component Certification and System Prediction* IEEE Computer Society, 2001 http://www.sei.cmu.edu/pacc/CBSE4-Proceedings.html

8. Ivica Crnkovic, Heinz Schmidt, Judith Stafford, and Kurt Wallnau. (eds) *Proceedings of the 5th ICSE Workshop on Component-Based Software Engineering: Benchmarks for Predictable Assembly* IEEE, 2002 http://www.sei.cmu.edu/pacc/CBSE5/CBSE5-Proceedings.html

9. Ivica Crnkovic, Heinz Schmidt, Judith Stafford, and Kurt Wallnau. (eds) Proceedings of the 6th ICSE Workshop on Component-Based Software Engineering: Automated Reasoning and Prediction IEEE, 2003

10. Eberhard von Faber. On the methodology for composite smart-card evaluation. In Proceedings of the 3rd Common Criteria conference, 2002 Describes issues in certifying smartcards according to the Common Criteria, and proposes a methodology where each of the several manufacturers involved contribute towards the overall evaluation.

11. Food and Drug Administration. *General Principles of Software Validation; Final Guidance for Industry and FDA Staff.* January 11, 2004. http://www.fda.gov/cdrh/comp/guidance/938.html This guidance outlines general validation principles that the Food and Drug Administration (FDA) considers to be applicable to the validation of medical device software or the validation of software used

to design, develop, or manufacture medical devices. Discusses composite systems, and places responsibility fully on the developer of the medical device.

12. Johan Fredriksson, Jerker Hammarberg, Joel Huselius, John Hakansson, Ola Larses, Markus Lindgren, Goran Mustapic, Anders Moller, Mikael Nolin, Thomas Nolte, Jonas Norberg, Dag Nystrom, Aleksandra Tesanovic, and Mikael Akerholm. *Component Based Software Engineering for Embedded Systems: A literature survey*. MRTC-Report no. 102, June 2003 http://www.mrtc.mdh.se/publications/0578.pdf This paper summarises and discusses a large number of papers on this topic by the authors. The CC and evaluation or certification are not explicitly discussed. The reviewed papers discuss many of the pertinent concepts in component-based software engineering. One paper describes experience with Boeing's Bold Stroke architecture, which defines an architecture for flight-control systems using components, some of them COTS.

13. Shari Galitzer, CygnaCom Solutions *Engineered Composition: the Results from Exploring an Approach* August 2003 Galitzer identifies several problem areas in the evaluation of composite systems, then suggests the use of OO techniques, specifically frameworks, design patterns, and aspect-oriented programming, to make composite evaluations easier.

14. Shari Galitzer, Cygnacom. *Introducing Engineered Composition (EC): An Approach for Extending the Common Criteria to Better Support Composing Systems.* September 2003 www.acsac.org/waepssd/papers/07-galitzer.pdf

15. Dimitra Giannakopoulou, Gary T. Leavens, and Murali Sitaraman (editors) *SAVCBS 2001 Proceedings: Specification and Verification of Component-Based Systems Workshop at OOPSLA 2001*. Technical report 01-09a, Department of Computer Science, Iowa State University, Nov 2001. http://www.cs.iastate.edu/~leavens/SAVCBS/2001/index.html

16. Dimitra Giannakopoulou, Gary T. Leavens, and Murali Sitaraman (editors) *SAVCBS 2003: Specification and Verification of Component-Based Systems* Technical Report 03-11, Department of Computer Science, Iowa State University, 2003. http://www.cs.iastate.edu/~leavens/SAVCBS/2003/index.shtml

17. J. Han and Y. Zheng. Security Characterisation and integrity assurance for software components and component-based systems, In *Proceedings of 1998 Australiasian Workshop on Software Architectures, Melbourne, Australia*, pages 83–89, 1998. Abstract: Software systems are increasingly assembled from components that are developed by and purchased from third parties, for technical and economic gains. In such component based software development, the functionality and quality-of-service attributes of the software components should be clearly and adequately specified (or packaged) through their interfaces, so that the characteristics of the systems assembled from the components can be analysed relative to the system requirements. The approach is partially based on the Common Criteria for Information Technology

18. Mr Naohisa Ichihara, NTTDATA Corporation, Japan. *A New Methodology for Composite Smartcard Evaluation.* In Proceedings of the Fourth Common Criteria conference, 2003. Describes the construction of a “composite ST” derived from the ST-lite of the underlying

hardware and the PP of the software. The PP is extended in various ways. Elements of the ST-lite may be modified, removed, reused, or "set-off."

19. Joint Interpretation Library. *ETR-lite for composition*. March 2002 www.cesg.gov.uk/site/iacs/itsec/media/joint-int-lib/JIL-ETR-lite-for-composition-V1-0.pdf www.commoncriteriaportal.org/public/files/2002-08-003.pdf. Defines documentation requirements intermediate between an ETR and a CR, intended to give enough information for evaluation of a TOE using the product as a component, without revealing proprietary information.

20. Joint Interpretation Library. *ST-lite*. July 2002 http://www.cesg.gov.uk/site/iacs/itsec/media/joint-int-lib/JIL-ST-lite-V1-1.pdf Defines a reduced form of ST for use in evaluating systems using the TOE as a component.

21. P.A. Karger and H. Kurth. *Increased information flow needs for high-assurance composite evaluations.* In Proceedings of the Second International Information Assurance Workshop (IWIA 2004), pp. 129-140, IEEE Computer Society, May 2004. Argues that the information requirements described in the ETR-lite are inadequate for EAL levels 5 and above, and proposes additional requirements for the exchange of information between hardware and software designers. Uses many anecdotes to illustrate his points. Notes that the very notion of composite evaluation is controversial and believed by many to be impossible.

22. Jim Krodel *Commercial Off-The-Shelf (COTS) Avionics Software Study.* U.S. Department of Transportation, Federal Aviation Administration, Office of Aviation Research report DOT/FAA/AR-01/26, May 2001 http://research.faa.gov/aar/tech/docs/techreport/01-26.pdf This report takes a snapshot of portions of industry domains related to safety. Avionics, nuclear, medical, space, and elevator domain information is surveyed. Key industry COTS components are identified and potential alternate methods for verifying a COTS component's applicability to the avionics domain application are studied.

23. A. Magar. Report on the Enhanced Windows-based Warning Terms Separation Proof-of-Concept (POC) Demonstrator. Technical Report DRD-006, Cinnabar Networks, Inc. April 2004

24. A. Magar. Report on Secure Access Management Proof-of-Concept (SAMPOC) II with Identity Management. Technical Report DRD-009, Cinnabar Networks, Inc. June 2004

25. Heiko Mantel. *On the Composition of Secure Systems.* In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, pp. 88-101, IEEE Computer Society, May 2002 This paper defines a framework for expressing information flow properties, based on a flow graph and six different primitive security predicates, that can be combined in various ways. The framework allows many known properties, e.g., noninterference, forward correctability, separability) to be expressed. A general "zipping" lemma is proved, which allows many composability results to be derived (most of them already known, but now with a simpler proof). Some "emergent" properties are also shown. This generalizes and extends the work on information flow started by McCullough, Sutherland, and others.

26. B. Maxey. *Integrating COTS in Safety Critical Systems Using RTCA/DO-178B Guidelines.* Presented at the 2nd International Conference on COTS-Based Software Systems. This paper examines the usage of commercial off the shelf (COTS) software embedded in sensor products for avionics applications. Usage of the guidelines of RTCA/DO-178B including consideration of independence, software criticality level and structural coverage are addressed. A comparison is made between development considerations for implementation of different software safety criticality levels.

27. Daryl McCullough Specifications for Multi-Level Security and a Hook-up Property. In *Proceedings of the 1987 Symposium on Security and Privacy*, pp. 161–166, April 1987.

28. Daryl McCullough. Noninterference and the composability of security properties. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 177–186, May 1988.

29. National Computer Security Center. *Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG-021, Version 1, April 1991. The Trusted Database Management System Interpretation extends the evaluation classes of the Trusted Computer System Evaluation Criteria to trusted applications in general, and database management systems in particular.

30. National Computer Security Center. *Trusted Network Interpretation of Trusted Computer System Evaluation Criteria*, NCSC-TG-005, Version 1, 31 July 1987. This document is an elaboration of the "orange book" evaluation criteria for networked systems. For the most part, the networked system is considered as a single evaluated system. An introduction discusses general issues. The "cascade problem" is identified and some guidance on its avoidance are given.

31. Peter Neuman. *Principled Assuredly Trustworthy Composable Architectures Final Report.* To appear 2004 www.csl.sri.com/users/neumann/chats4.html This report characterizes problems in, and approaches to, attaining computer system and network architectures, with the overall goal of being better able to develop and more rapidly configure highly trustworthy systems and networks able to satisfy critical requirements. Chapter 3, "Realistic Composability," discusses technical issues relating to composite systems, and notes mechanisms that can be used to enhance the reliability of systems (so that the security property is a consequence of the architecture rather that the individual components). Chapter 6 of the report discusses assurance. The report is sweeping in scope, with well over 300 references. The report concludes with a list of research recommendations; most relevant to our topic is research into "pervasive integrated assurance."

32. George Romanski, Verocel, Inc. *The Challenges of Software Certification* STSC CrossTalk, Sept 2001 http://www.stsc.hill.af.mil/crosstalk/2001/09/romanski.html

33. RTCA, Inc. Software Considerations in Airborne Systems and Equipment Certification. DO-178B/ED-12B. December 1992.

34. Roger R. Schell. Information Security: Science, pseudoscience, and flying pigs. In Proceedings of the 17th Annual Security Applications Conference, pp. 205-216, December 2001. www.asac.org/invited-essay/essays/2001-schell.pdf

35. Sha, Goodenough, Pollak *Simplex Architecture: Meeting the Challenges of Using COTS in High-Reliability Systems* STSC CrossTalk, Apr 1998 http://www.stsc.hill.af.mil/crosstalk/1998/04/simplex.asp Discusses software architecture for fault-tolerance, allowing COTS to be used in a high-reliability system. Not specifically about evaluation.

36. Sean Smith, Ron Perez, Steve Weingart, and Vernon Austel. Validating a High-Performance, Programmable Secure Coprocessor. *Proceedings of the 22nd National Information Systems Security Conference.* October 1999. http://csrc.ncsl.nist.gov/nissc/1999/proceeding/papers/p16.pdf

37. A M. Tinto. *The design and evaluation of INFOSEC systems: The computer security contribution to the composition discussion.* C Technical report 32-92, National Computer Security Center, June 1992. http://www.radium.ncsc.mil/tpep/library/rainbow/C-TR-32-92.pdf This paper discusses how the composability problem is addressed in the Trusted Computer Systems Evaluation Criteria (TCSEC); the Network Interpretation (TNI), and the DataBase Interpretation (TDI). These three documents, taken together, define a framework for the evaluation of different types of composite systems and different decomposition architectures.

38. Gary Veccellio and William M. Thomas *Issues in the Assurance of Component-Based Software* 2000 International Workshop on Component-Based Software Engineering http://www.sei.cmu.edu/pacc/cbse2000/papers/19/19.pdf This paper reviews standards for COTS software in critical applications, identifies new challenges associated with component based COTS software, and suggests some static analysis approaches to help address the these challenges.

39. Jeffrey Voas. *Composing Software Component "ilities."* IEEE Software, July/August 2001. Voas discusses non-functional properties (e.g., reliability) of composites, and argues that none of the most important such properties are easily composed, and that tools and techniques for the analysis of composites are not available.

40. Simon R.Wiseman and Lt.Col. Colin J. Whittaker. *A New Strategy for COTS in Certified Systems.* 1998. http://security.isu.edu/pdf/newcots.pdf Describes U.K. MOD's "emerging strategy for Infosec." A prototype system, "Purple Penelope," was developed to validate the approach. That experiment was quite successful. Some components are certified to ITSEC E3; each runs at a single level while the system as a whole is multi-level.

41. S. Zeber and A. Magar. *Managing Identity and Access in the Defence Environment.* Technical memorandum TM 2002-056, DRDC Ottawa , April 2002

42. George Blakley and Helmut Kurth. *Evaluation of an Access Control Framework*. In the 4th International Common Criteria Conference.

List of symbols and abbreviations

CC	Common Criteria. An international standard for the evaluation of security properties of IT systems.
CEM	(from CC) Common Evaluation Methodology. A description of the CC evaluation process.
COTS	Commercial off-the-shelf software.
CR	Certification Report (from CC).
DND	Department of National Defence
EAL	Evaluation Assurance Level.
ETR	Evaluation Technical Report.
ITSEC	Information Technology Security Evaluation Criteria.
JIL	Joint Interpretation Library
OR	Observation Report.
PP	Protection profile
TCSEC	Department of Defense Trusted Computer System Evaluation Criteria (DOD-5200.28-STD).
TNI	Trusted Network Interpretation, NSA NCSC TG-005 “red book” in the rainbow series.
TOE	Target of evaluation, from CC. The TOE is the part of the system being evaluated.
TSF	TOE security functions, from CC. The elements of the TOE that must be relied upon for the enforcement of the TSP
TSFI	TSF Interface, from CC.
ST	Security target, from CC

Glossary

Common Criteria	An international standard for the evaluation of security properties of IT systems.
Common Evaluation Methodology	A description of the CC evaluation process.
Certification	(From the CC) The final acceptance of an evaluation verdict. The evaluation results are subjected to independent inspection by a certifying body before being accepted.
Class	(From the CC) A general grouping of security requirements. A class is made up of families.
Component	(From the CC) A specific set of security requirements defined in the CC (e.g., ADV_SPM.3).
Component	(Common usage) A part or subsystem.
Certification Report	(From the CC) A brief description of a successful evaluation.
Emergent	A property arising from a combination of systems rather than from the systems themselves.
ETR-Lite	Abbreviated version of an ETR with sensitive or proprietary infocmation removed, from Joint Interpretation Library. See [19].
Evaluation Assurance Level	(From the CC) Each EAL defines a set of assurance components that must be met. EAL 1 is the least stringent, and EAL 7 the most.
Evaluation	(From the CC) Assessment of whether the TOE satisfies the requirements and objectives of the ST. An evaluation results in an ETR, any number of ORs, and, if successful, in a CR. (The CC also defines evaluation for PPs and STs.)
Evaluation Technical Report	(From the CC) A report written by an evaluator, providing technical justification for the evaluation verdict.
Family	(From the CC) Grouping of security requirements sharing security objectives. Specific families are defined in the CC, for example, ADV_SPM. The members of a family are components.
Information Technology	Developed in the early 1990s, now recognized through Europe.

Security Evaluation Criteria	The ITSEC was one of several standards used in developing the Common Criteria.
Joint Interpretation Library	Collection of documents produced by the Joint Interpretation Library Working Group, representing France, Germany, the Netherlands and the UK.
Module	(From the CC) Subsystems are composed of modules (which is left undefined by the CC). A module is meant to be a low level element of a system.
Observation Report	(From the CC) A mechanism allowing an evaluator to request clarification or identify a problem with an evaluation.
Protection Profile	(From the CC) The protection profile defines an implementation-independent set of security requirements and security objectives for a class of systems.
Security target	(From the CC) The ST defines the security requirements and security objectives for a TOE, and defines the mechanisms offered by the TOE to meet the requirements.
ST-Lite	An abbreviated version of a ST, defined in the Joint Interpretation Library. See [20].
Subsystem	(From the CC) A major architectural element of a system. Systems are composed of subsystems and subsystems are composed of modules.
Trusted Computer System Evaluation Criteria	U.S. Department of Defense(DOD-5200.28-STD). CSC-STD-001-83 “Orange Book” in the rainbow series.
Trusted Network Interpretation	NSA NCSC TG-005 “red book” in the rainbow series.
Target of evaluation	(From the CC) The TOE is the part of the system being evaluated.
TOE security functions	(From the CC) The elements of the TOE that must be relied upon for the enforcement of the TSP

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

ORA Canada
P.O. Box 46005, 2339 Ogilvie Road,
Ottawa, ON K1J 8M6

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Review of the Composability Problem for System Evaluation (U)

4. AUTHORS (Last name, first name, middle initial)

Craigen, D., Saaltink, M.

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (total cited in document)
November 2004	50	42

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DRDC Ottawa/IO Section
3701 Carling Avenue
Ottawa K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)
15BF27	W7714-4-8934

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)
TR-2004-5601-2	DRDC Ottawa CR 2004-196

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(x) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Full Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This report presents the results of a review of the security evaluation of composite systems to determine the current status of this topic, especially with respect to work that has been done in this area and tools and techniques that have been developed. Recommendations on how security evaluation could be applied to the DRDC/NIO Secure Access Manager are also included.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Assurnace Continuity, Certification, Common Criteria, Composability, Composite Systems, COTS, Evaluation, Evaluation Kits, Regulatory Data Banks, Secure Access Manager, Security Evaluation, Standards,