

A Comparison of the S-vector Methodology to the Common Criteria
August 2004

Janine Spears, Russell Barton, William Hery



The Penn State eBusiness Research Center

**401 Business Administration Building
University Park, PA 16802**

**Phone: 814.863.7575 Fax: 814.865.9119
Web: <http://www.ebrc.psu.edu>**

PENNSTATE

SMEAL College of Business Administration

Corporate Sponsors: IBM • UNISYS • XEROX • AT&T Wireless • Delphi Ventures • SAP AG • CIGNA • TYCO • HP

Abstract

This paper compares the S-vector methodology to the Common Criteria in order to identify distinctions between these two approaches to web application security. The paper is organized as follows. The first section provides an introduction to the Common Criteria. Second, a comparison between the S-vector and Common Criteria methodologies is made. Third, suggestions for extending the Common Criteria to accomplish the goals of the S-vector methodology are provided. Finally, questions that impact future S-vector research, and therefore require additional discussion, are presented.

Author Biographies

Russell Barton is Associate Dean for Research and Ph.D./M.S. Programs and Professor of Supply Chain and Information Systems in the Mary Jean and Frank P. Smeal College of Business at Penn State. He received a B.S. in Electrical Engineering from Princeton University and M.S. and Ph.D. degrees in Operations Research from Cornell University. After ten years in industry, most of that time at RCA's David Sarnoff Research Center, he returned to academia, teaching at Cornell from 1987-1990 and then joining Penn State. At Penn State, he has received more than \$1.5 million for research in statistical process control, the design of experiments, and computer simulation, and has worked with industry sponsors including Boeing, Fluke, Ford, General Motors, Hewlett-Packard, Intel, Lucent, New-Holland and Xerox. Dr. Barton has taught short courses in concurrent engineering, design for manufacturing, and the graphical design and analysis of experiments. He has recently completed a text, Graphical Methods for the Design of Experiments, published by Springer-Verlag. He has worked to increase the practice component of engineering education at Penn State, and has received one national, one university, three college and two departmental teaching and curriculum development awards since 1990. He is a senior member of IEEE and IIE, and a member of ASQ and INFORMS. He has held a number of service positions in the INFORMS Simulation Society and is currently president. He is Program Chair for the 2007 Winter Simulation Conference.

William Hery is an eBRC Research Center Fellow in the Mary Jean and Frank P. Smeal College of Business at Penn State. Dr. Hery has more than 25 years of experience in research and development in industrial R&D environments, and 8 years of university level teaching of mathematics and computer science. The focus of his research has been the application of computer science, operations research, and mathematics to the understanding of large systems in a variety of contexts. During the most recent eight years he has focused on information and communications security in distributed processing environments. He is currently helping to define and build the Cybersecurity component for the Polytechnic University's Urban Security Initiative. He is also an advisor to a government agency on a homeland defense project. In 2001, Dr. Hery retired from Lucent Technologies Bell Labs, where he was a Distinguished Member of Technical Staff in the Government Communications Lab. During most of his 18 years at Bell Labs, he led R&D projects related to the Defense and Intelligence communities, focusing the technologies of Bell Labs' forward looking research on the special needs of the government.

Janine Spears is a doctoral candidate in the department of Supply Chain and Information Systems in the Mary Jean and Frank P. Smeal College of Business at Penn State. She received a B.S. in Computer Information Systems from California State University at Los Angeles and an M.B.A. from Case Western Reserve University. Her research interest is in the management of information security. Prior to joining Penn State, Janine worked as a Business Systems Analyst at Twentieth Century Fox, Sony Pictures Entertainment, AST Computers, and the Jet Propulsion Laboratory. She also taught courses in information systems at Cuyahoga Community College, Santa Monica College, and California State University at Los Angeles.

1. Introduction to the Common Criteria

This section describes the Common Criteria, to include a brief description of its history, the organization of the standard, and the major sections within the standard. The section concludes with a description of how an evaluation is accomplished using the Common Criteria.

1.1 What is the Common Criteria?

The Common Criteria for Information Technology Security Evaluation, commonly referred to as the Common Criteria (CC), is used to specify and evaluate security requirements of a targeted system. The CC is an international standard that was derived from a collection of security standards from Canada, England, France, Germany, and the United States. CC version 1.0 was completed in 1996. The CC became ISO standard 15408 in 1999. The current version, CC 2.1, was developed in August 1999. Since the CC's inception, several other countries have joined the consortium of nations that recognize CC certifications, bringing the total to approximately 19 countries. This means that the certification of a product in one country is recognized and accepted in all other member countries.

The system being evaluated under the CC, referred to as the Target of Evaluation (TOE) may include, but is not limited to, an operating system, database, network, or application. The CC provides a list of functional security requirements that may be applied to a TOE. The CC also provides assurance requirements to ensure that requirements are implemented to a desired level of scope, rigor, and depth.

1.2 Organization of the Common Criteria

CC version 2.1 is defined in a three-part document, totaling 618 pages, and includes an introduction (Part 1), functional security requirements (Part 2), and assurance requirements (Part 3). The introduction provided in Part 1 includes the CC's terminology, goals, objectives, and format.

Part 2 contains a set of *functional requirements* that may be applied to the TOE (i.e., the system being evaluated). Functional requirements are grouped by category. Each category is referred to as a class. Within each class, functional requirements are organized (grouped) hierarchically into families, and then components. There are eleven classes of requirements in CC Part 2, and include:

- Security audit
- Communication (deals with Non-repudiation)
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Privacy
- Protection of the TOE security functions
- Resource utilization
- TOE Access
- Trusted path

Part 3 of the CC contains a set *assurance requirements* that may be used to define the scope, depth, and rigor of a security evaluation. Assurance requirements are grouped by class, and then further grouped into families and components. The eight assurance classes included in CC Part 3 are:

- Configuration management
- Delivery and operation
- Guidance documents
- Life-cycle support
- Tests
- Vulnerability assessment
- Maintenance of assurance

The CC defines a structure for creating *Protection Profiles* (PP). A PP is a document that is typically created by a user to specify the security requirements needed by a system to ensure an acceptable level of security. Requirements specified in a PP may come from Parts 2 or 3 of the CC, or they may be derived from other sources. A PP enables a user group to communicate desired security requirements and may be directed at a specific TOE (e.g., a specific web application), or a type of TOE (e.g., web applications in general). The PP concept has been developed to support the definition of functional standards as an aid to procurement specifications and is independent of a specific implementation (Hayes).

The CC defines a structure for creating *Security Targets* (ST). An ST is a document that specifies the security requirements against which the TOE will be evaluated. An ST is associated with a specific TOE and may be associated with zero, one, or multiple Protection Profiles. In other words, the ST states what security is actually provided by the TOE, and this may or may not include requirements specified in a PP. When a system is evaluated using the CC, it is evaluated against the ST defined for that system.

The CC defines a measurement scale called *Evaluation Assurance Levels* (EAL). There are seven EALs (1-7), with each successive number indicating higher levels of assurance. An EAL indicates the level of testing a product has undergone; the more rigorous and formalized the testing process, the higher the assurance. The international community recognizes EAL1 through EAL4; however the criteria for levels 5-7 have not reached international agreement and therefore are not recognized internationally. This means that the highest level a product can be certified and have that certification internationally recognized is at EAL4. The CC provides a list of criteria to be met for each EAL. Alternatively, an organization may define its own set of criteria per EAL, in which case the EAL is said to be “extended.”

Although the CC provides a list of functional and assurance requirements, usage of these requirements within a PP or ST is optional. An organization may decide to employ security requirements derived externally from the CC, while using the CC as a framework for developing a PP or ST, or for conducting a security evaluation using the CC’s EAL structure.

1.3 Product Certification using the Common Criteria

A third-party evaluator must first certify that the Security Target (ST) is complete and in compliance with the CC. If a PP is used, it must be certified for its compliance with the CC. Certification of a ST or PP ensures the document is realistic and complete. An organization determines the desired EAL against which the TOE is to be certified. The organization then ensures that it meets the criteria defined for the desired EAL. A third-party evaluator evaluates the TOE against the ST in a certified laboratory (external to the organization seeking certification), and awards the accomplished EAL.

2. Comparison of the S-vector and Common Criteria Methodologies

This section identifies similarities and differences between the S-vector methodology (SV) and the Common Criteria (CC). Before making this comparison, a list of the components defined for SV is helpful. Based on the S-vector Full NSF Proposal (Barton et al., 2004) and the S-vector Strawman (Hery & Liu, 2003), SV has been defined to include the following:

- 13 technical components (corresponding to CC functional components)
- procedural components (management-related)
- structural components (related to program source code of web application)
- a requirements vector (loosely corresponding to CC's security target)
- an application scoring vector

2.1 Similarities between the S-vector Methodology and the Common Criteria

The structure of the S-vector methodology loosely corresponds with the CC structure, but is not as detailed. For example, S-vector components correspond with CC components. The S-vector methodology uses a requirements vector to record targeted security elements; the CC uses a security target (ST) for the same purpose. The CC groups security components by Class → Family → Components. Although the S-vector methodology has not defined a means of grouping related technical, structural, or procedural components, it is conceivable that such a grouping would be needed if an S-vector were to contain a large number of components (currently there are only 13 technical and fewer structural and procedural components defined).

One could argue that SV differs from CC in its more simplistic approach to security assessment (e.g., smaller number of security components to evaluate, less rigorous testing procedures, etc.). However, the CC is designed to provide flexibility in the security requirements being evaluated, the definition of EALs, and the assurance methodology applied. SV may choose to reduce scope (e.g., web applications only), depth (e.g., only certain aspects of the web application), and rigor (e.g., not using 3rd party labs for appraisals). However, reduced scope, depth, and rigor do not distinguish SV from CC; this reduction in complexity would be interpreted as an alternative method of applying the CC.

2.2 Differences between the S-vector Methodology and the Common Criteria

SV provides a method for scoring the implementation of a security component (i.e., an algorithm), while the CC does not. The CC provides a measurement scale via the EALs, and a framework for documenting assurance requirements.

Based on my initial interpretation of the CC, only the final EAL score for the entire product (TOE) is intuitive for users; the CC is not used to make comparisons among CC security classes or families assigned to a TOE in order to prioritize security enhancements. This is a key goal of the S-vector methodology.

The CC does not provide criteria for evaluating procedural components (referred to as administrative measures in CC terms). Conversely, SV will include scoring criteria for procedural components.

The CC primarily evaluates security from a product's design through its release. As described in the NSF proposal, SV enables an organization to prioritize security *enhancements*, which implies SV assessments may typically be conducted on web applications already in operation.

The S-vector methodology aims to provide a low-cost and easy-to-use security assessment tool for web applications, which is not the goal of the CC. The Common Criteria (CC) is a detailed evaluation process typically used for commercial, security-related, products and for key mission critical federal government applications. Commercial products that have been CC certified are listed on an NIST web site (http://www.niap.nist.gov/cc-scheme/vpl/vpl_type.html), and include products such as firewalls, anti-virus software, and database management systems. (Note that although the CC may be applied to web applications, no web applications are listed as being certified.) These commercial products generally require more rigorous security testing than non-commercial applications. One reason for this is that commercial software vendors use CC certification as a marketing tool (*in addition to* practical security testing) by assuring potential customers the certified product has been rigorously tested. As such, certification is used to influence potential customers' purchasing decisions. Commercial vendors can hand down costly certifications to the end customer through a product's pricing. Although targeted SV users are concerned with the security of their web applications, they are not selling their applications for commercial use and do not require the additional assurances required of commercial products.

3. How the S-vector Methodology may be used to extend the Common Criteria

The CC document states that it may be used to evaluate web applications. However, there are no guidelines on which components of the CC would most effectively address web application security specifically. Secondly, the CC is more commonly used for commercial security products, such as firewall software. SV could extend the CC by packaging a subset of CC for use in evaluating web applications specifically. This packaging could include security components, a default security target that could be customizable, and EAL definitions specific to web applications.

In meeting the goal of SV to provide a low-cost assessment tool, SV can extend the CC by defining a subset of the CC that provides acceptable security, while excluding portions of the more rigorous testing required of commercial products. Defining a low-cost, effective implementation of the CC would *provide broader appeal* to the general population by enabling more organizations to evaluate web application security. This broader pool of organizations would include non-profit organizations and small businesses, which also includes traditionally disadvantaged populations (in reference to NSF feedback). These organizations typically will not have the financial resources to undertake the rigorous testing outlined in the full CC; nor will they typically have the resources to analyze and define a smaller, effective subset of the CC to apply to their web applications. This paired down version of the CC would also define a means for evaluation that does not require the use of third-party evaluators and/or external, certified laboratories.

The CC provides a list of functional requirements that an organization may want to evaluate during a security assessment. These functional requirements are technical in nature. SV can extend the CC by including administrative components (termed "procedural" in SV documentation). SV can include administrative components from ISO 17799 and SSE-CMM, while making necessary modifications to the assurance requirements outlined in CC in order to effectively evaluate these components. Joining these three leading security standards would provide broad appeal.

SV can extend the CC by providing a user-friendly interface in which to report assurance testing results. This interface would provide a means for comparing classes of security within a single web application

and for comparing security between multiple web applications. This interface could possibly enable users to build customized reports.

4. Questions requiring further discussion and research

What is the *specific scope and depth of SV*? What is included in web application security? Does the security of web applications as addressed by SV include the databases or network used by a web application? Are the 13 high-level technical components listed in the Strawman to be interpreted as a comprehensive set of SV technical components, or is the intention to greatly expand upon this set? What level of assurance is SV attempting to provide? Will an organization using SV to assess a web application's security use SV as the sole tool for determining if an application is secure? Answering these questions would help to determine the scope and depth that SV must provide.

What is meant by a *low-cost assessment tool, and how can this be achieved*? Related to scope, what tradeoffs are we willing to make in order to produce a low-cost, easy-to-use, and effective security assessment tool? The specifics of these tradeoffs must be defined. Keeping in mind the clearly defined scope of SV, how can SV provide a low-cost *and* effective assessment? What level of technical expertise is needed for an organization to effectively use SV? If a greater emphasis will be placed on automated tools for assessing SV components, how can SV be implemented at a low cost; system software tools are typically very expensive to purchase, setup, and implement.

For technical and procedural aspects that are peripheral, but integral to web application security being addressed by SV, **what specific assumptions are being made?** For example, does SV assume that a risk analysis has been performed, that network security is covered by a separate assessment method, etc.? Those aspects of security that are not addressed by SV, but that impact web application security, must be explicitly identified so that an organization can determine how it should address these aspects. Without a clear, mutual understanding of these aspects, the perceived effectiveness of an SV implementation could be compromised.

References

Barton, R., Hery, W., Liu, P., A Scoring Vector for Managing Web Application Security, NSF Full Proposal, Mar 2004.

Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 2.1, CCIMB-99-031, Aug 1999, 56 pages.

Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements, Version 2.1, CCIMB-99-032, Aug 1999, 354 pages.

Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements, Version 2.1, CCIMB-99-033, Aug 1999, 208 pages.

Hayes, K., Common Criteria – A World Wide Choice, <http://www.itsecurity.com/papers/88.htm>.

Hery, W., Liu, P., Strawman S-vector Structure, 2003.