

Synopsis of Evaluating Security Controls Based on Key Performance Indicators and Stakeholder Mission Value

Robert K. Abercrombie
Frederick T. Sheldon
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6418 USA
abercrombie@ornl.gov sheldonft@ornl.gov

Ali Mili
College of Computing Sciences
New Jersey Institute of Technology
Newark, NJ 07102-1982 USA
mili@cis.njit.edu

Abstract

Information security continues to evolve in response to disruptive changes with a persistent focus on information-centric controls and a healthy debate about balancing endpoint and network protection, with the goal of improved enterprise and business risk management. Economic uncertainty, intensively collaborative work styles, virtualization, increased outsourcing and ongoing compliance pressures require careful consideration and adaptation of a balanced approach. The Cyberspace Security Econometrics System (CSES) provides a measure of reliability, security and safety of a system that accounts for the criticality of each requirement as a function of one or more stakeholders' interests in that requirement. For a given stakeholder, CSES reflects the variance that may exist among the stakes one attaches to meeting each requirement. This paper summarizes the basis, objectives and capabilities for the CSES including inputs/outputs as well as the structural underpinnings.

1. Introduction

The lack of sound and practical security metrics is severely hampering progress in the development of secure systems. The Cyberspace Security Econometrics System (CSES) offers the following advantages over traditional measurement systems: (1) CSES reflects the variances that exist among different stakeholders of the same system. Different stakeholders will typically attach different stakes to the same requirement or service (e.g., a service may be provided by an information technology system or

process control system, etc.). (2) For a given stakeholder, CSES reflects the variance that may exist among the stakes one attaches to meeting each requirement. The same stakeholder may attach different stakes to satisfying different requirements within the overall system specification. (3) For a given compound specification (e.g., combination(s) of commercial off the shelf software and/or hardware), CSES reflects the variance that may exist among the levels of verification and validation (i.e., certification) performed on components of the specification. The certification activity may produce higher levels of assurance across different components of the specification than others.

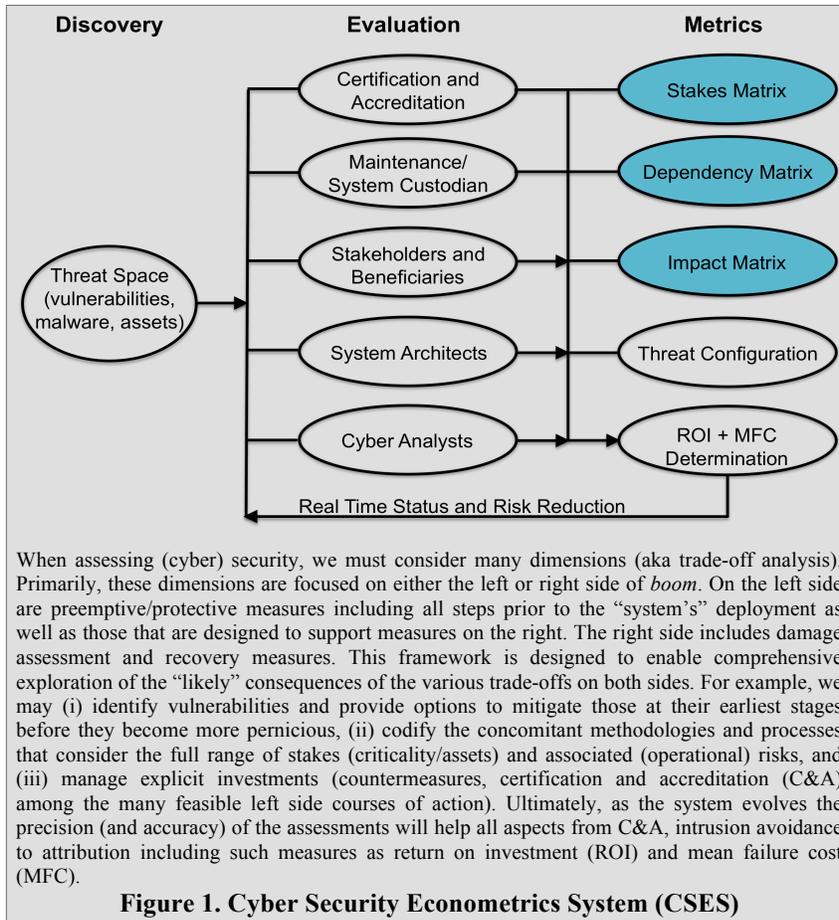
1.1. Motivation

Traditionally, the verification and validation (V&V) effort is charged uniformly on all stakeholders. With the quantification infrastructure that has been previously introduced to compute Mean Failure Cost (MFC), we can employ a scheme where the cost of any V&V effort is charged on the stakeholders according to what they stand to lose or gain [1]. Hence if a particular V&V effort is aimed at improving the level of confidence that refines a component (i.e., that implements a service and/or satisfies a requirement), then stakeholders are charged according to the stake they have in satisfying said requirement. CSES also introduces and combines such measures as verification costs which considers the fact that it may be easier to verify a system against one requirement component than against another. Such costs depend on the system, the requirement and the selected verification method.

1.2. Related Works

The proposed metric is consistent with the spirit of Value Based Software Engineering [2-4]. Whereas the

The submitted manuscript has been authored by a contractor of the U.S. Government under contract DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.



Mean Time to Failure (MTTF) is an abstract quantity that reflects the failure rate of a system, the MFC quantifies the impact of failures by providing a failure cost per unit of time. This cost must be balanced against the benefit of operating the system for the same unit of time, to determine the desirability of operating

The history of Value Based Software Engineering, to date, is well documented [5]. Originally, software engineering only dealt with the technical challenges. This has changed over the years, especially in industry projects, when value was introduced to aid in the decision making process [6]. Historically, software engineering practice and research has been conducted in a value-neutral setting [3]. This approach led to an underestimation of the need to align the incentives of success-critical stakeholders [7]. Introduction of financially responsible approach to requirements prioritization enhanced value created potential [8]. Many evaluation approaches analyze costs, benefits and risks associated with Information Technology in general (e.g., cost-oriented approaches, multi-dimensional approaches, market-oriented approaches, strategy-oriented approaches, customer-oriented approaches and process-oriented approaches) [9].

systems [13].

2. Concepts and Assumptions

Figure 1 shows essential input/output components and phases (i.e., discovery, evaluation and metrics) including data collection/analysis and consisting of the following entities [14, 15]:

- **System Stakeholders** are any person or organization that has a stake in the operation of the system (i.e., users, operators of the system, hosts of the systems, etc.).
- **Security Specification** used in the same way that correctness is a relative attribute (a system is correct with respect to its functional specification) and refers to a representation of the security attributes that a system must satisfy to be deemed secure.
- **Security Requirement** used in the same way that a complex functional specification is typically composed of simpler components (representing elementary functional properties), and is composed of simpler security requirements.

Finally, controlling-oriented approaches unified the concepts of earned valued management and target costs. This was in turn influenced by Value Based Software Engineering [9]. Tracing value-base requirements and their impact has always been a challenge. A case study on value-based requirements tracing, that systematically supported project managers in tailoring requirements tracing precision and effort based on parameters stakeholder value, requirements risk/volatility, and tracking costs, illustrated this [10]. Other studies describe techniques required for distributed priority ranking of strategic requirements for information systems in economic organization [11]. Within the last few years, studies have made attempts to understand the stakeholder view of quality [12]. Recently the mapping from SSE-CMM process areas to the patient-centered healthcare domain has the potential to establish a set of metrics to assess security risks for patient-centered healthcare

- **Mean Failure Cost (MFC)** used in the operational sense because the lack of security within the system may cause damage, in terms of lost productivity, lost business, lost data, resulting in security violations. We represent this loss by a random variable, and define MFC as the mean of this random variable [1]. As discussed further, this quantity is not intrinsic to the system, but varies by stakeholder [16].

3. Step-Wise Process of CSES

To estimate the MFC of a system for a set of stakeholders, we initially identify and then maintain (from the discovery phase) the following information: (1) the set of stakeholders of the system, and (2) the set of security specifications and thus security requirements that are to be satisfied by the system. (3) For each stakeholder and each security requirement, the stake that the selected stakeholder attaches to the selected service (or conversely, the cost that the stakeholder incurs if the service is disrupted). This information is provided by stakeholders. (4) For each component of a specific security requirement, the likelihood that the system provides that service as specified. This information is computed in light of the V&V measures (inspection, verification, testing, security measures, firewalls, vulnerability removal, threat mitigation, etc) that the system has been subjected to. In particular, estimating the likelihood of delivering a service requires that we determine to what degree the components involved in delivering a service have been validated. Thus, following the CSES vertical process of the Metrics Engine proceeds in three steps (applying Stake Estimation to generate the Stakes Matrix, Bayesian Analysis to generate the Dependency Matrix, and Threat Analysis to generate the Impact Matrix) by the subject matter experts as described in the vertical Evaluation Engine components [14, 15]. CSES encompasses not only failure costs but also mitigation costs, specifically verification costs. Once the basic matrices are populated, a baseline for the particular instantiation of the CSES is established and all changes to the baseline are maintained in a way that track the enterprise's evolution to provide near real-time assessments.

4. Definitive Instantiation

As shown in Figure 1, the system follows a defined process. The initial inputs (1) organization mission (and components thereof), (2) value of its objectives and assets if uninterrupted, and (3) the components of the enterprise system that support each mission component, are determined by stakeholders.

The stakeholder/customer, with assistance from Subject Matter Experts (SMEs), defines their criteria for evaluating their assets. For example, the criteria may include:

- Financial basis (e.g., operational cost of downtime per unit of time defined with hardware/software costs, HVAC, staffing, etc., versus profit); which is the quantitative measurement to be used within the CSES.
- Federal Information Security Management Act (FISMA) of 2002, customer derived value of assets per NIST 800-60, and/or FIPS 199/200 (February 2004, Standards for Security Categorization of Federal Information and Information Systems) dictated requirements.
- Customer defined requirements; acceptable and unacceptable impact levels against cost value related to Information Assurance tenets of confidentiality, availability and integrity may also be examined.

Variances exist among different stakeholders of the same system. Different stakeholders will attach different stakes to the same requirement or service (e.g., a service may be provided by an information technology system or process control system, etc.). For a given stakeholder, CSES will reflect the variance that may exist among the stakes the stakeholder attaches to meeting each requirement. The same stakeholder may attach different stakes to satisfying different requirements within the overall system specification. Once CSES is base lined and it evolves, compound specification (e.g., combination(s) of commercial off the shelf software and/or hardware) will become apparent. For a given compound specification, CSES will reflect the variance that may exist among the levels of V&V (i.e., certification) performed on components of the specification. The certification activity may produce higher levels of assurance across different components of the specification than others.

For each component of a specific security requirement, the likelihood that the system provides that service as specified. This information is computed in light of the V&V measures (inspection, verification, testing, security measures, firewalls, vulnerability removal, threat mitigation, etc) that the system has undergone. In particular, estimating the likelihood of delivering a service requires that we analyze to what degree the components that are involved in delivering this service have been validated.

4. Conclusions

The CSES process proceeds in three steps (Generation of Stakes Matrix, Dependency Matrix and

Threat Matrix). CSES encompasses not only failure costs but also mitigation costs, specifically verification costs. CSES provides:

- A framework for measuring the appropriate attributes that support the decisions necessary to (1) design security countermeasures, to choose between alternative security architectures, (2) respond to events such as intrusions or attacks and, (3) improve security (including reliability and safety) during both design and operational phases.
- A comprehensive basis for choosing courses of action that have the highest risk reduction return on investment (i.e., reduce the most risks for the lowest cost).

CSES and its underpinning rationale are (1) consistent with the spirit of Value Based Software Engineering and (2) comprehend the different organizational mission needs for all stakeholders. For example, CSES identifies information assurance controls and mitigation costs as an investment toward assuring mission success.

On the practical side, we need to find sample applications where deployment of the CSES with its associated MFC metric show usefulness and superiority, by providing a sound basis for analysis and decision-making. On the theoretical side, we need to develop the mathematical infrastructure that allows us to estimate or to approximate the MFC using failure costs and failure probabilities given (respectively) by stakeholders and engineers (V&V teams).

4. References

- [1] A. Mili and F. T. Sheldon, "Measuring Reliability as a Mean Failure Cost," in *Proceedings of the 10th IEEE High Assurance Systems Engineering Symposium*, 2007, pp. 403-404.
- [2] S. Biffi, A. Aurum, B. W. Boehm, H. Erdogmus, and P. Gruenbacher, "Value Based Software Engineering," Springer Verlag, 2006.
- [3] B. W. Boehm and L. Huang, "Value Based Software Engineering: A Case Study," *IEEE Computer*, vol. 36(3), March 2003.
- [4] B. W. Boehm and L. Huang, "Value Based Software Engineering: Reinventing Earned Value Monitoring and Control," *ACM Software Engineering Notes*, vol. 28(2), March 2003.
- [5] B. Boehm, "A View of 20th and 21st Century Software Engineering," in *Proceedings of the 28th International Conference on Software Engineering* Shanghai, China: ACM, 2006.
- [6] H. Omasreiter, "Balanced Decision Making in Software Engineering--General Thoughts and a Concrete Example from Industry," in *Proceedings of the First International Workshop on The Economics of Software and Computation*: IEEE Computer Society, 2007.
- [7] A. Egyed, S. Biffi, M. Heindl, and G. Paul, "A Value-Based Approach for Understanding Cost-benefit Trade-offs during Automated Software Traceability," in *Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering* Long Beach, California: ACM, 2005.
- [8] J. Cleland-Huang and M. Denne, "Financially Informed Requirements Prioritization," in *Proceedings of the 27th International Conference on Software Engineering* St. Louis, MO, USA: ACM, 2005.
- [9] B. Mutschler, J. Bumiller, and M. Reichert, "Designing an Economic-Driven Evaluation Framework for Process-Oriented Software Technologies," in *Proceedings of the 28th International Conference on Software Engineering* Shanghai, China: ACM, 2006.
- [10] M. Heindl and S. Biffi, "A Case Study on Value-Based Requirements Tracing," in *Proceedings of the 10th European Software Engineering Conference held jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering* Lisbon, Portugal: ACM, 2005.
- [11] A. Sobczak and D. M. Berry, "Distributed Priority Ranking of Strategic Preliminary Requirements for Management Information Systems in Economic Organizations," *Information and Software Technology*, vol. 49, pp. 960-984, 2007.
- [12] B. Boehm, S. Chulani, J. Verner, and B. Wong, "Sixth Workshop on Software Quality," in *Companion of the 30th International Conference on Software Engineering* Leipzig, Germany: ACM, 2008.
- [13] L. Huang, X. Bai, and S. Nair, "Developing a SSE-CMM-based Security Risk Assessment Process for Patient-Centered Healthcare Systems," in *Proceedings of the 6th International Workshop on Software Quality* Leipzig, Germany: ACM, 2008.
- [14] F. T. Sheldon, R. K. Abercrombie, and A. Mili, "Evaluating Security Controls Based on Key Performance Indicators and Stakeholder Mission," in *Proceedings of the 4th Annual Cyber Security and Information Intelligence Research Workshop* Oak Ridge, TN: ACM, 2008.
- [15] F. T. Sheldon, R. K. Abercrombie, and A. Mili, "Methodology for Evaluating Security Controls Based on Key Performance Indicators and Stakeholder Mission," in *Proceedings of 42nd Annual Hawaii International Conference on System Sciences*. vol. 42 Waikoloa, HI: IEEE, 2009.
- [16] A. Mili and F. T. Sheldon, "Challenging the Mean Time to Failure: Measuring Dependability as a Mean Failure Cost," in *Proceedings of 42nd Hawaii International Conference on System Sciences*. vol. 42 Waikoloa, HI: IEEE, 2009.