

# Measuring Software Security

Julia Allen

Since the mid 1990s, CERT has researched and created value-added processes, methods, practices, and tools for software survivability, software assurance, and building security into software throughout its development life cycle. In recent years, the research community has increasingly contributed to the body of knowledge about software assurance and software security metrics.

Unfortunately, the security community often conflates information security metrics and software security metrics, which in fact are quite distinct. Efforts to identify meaningful information and operational security metrics have been ongoing for some time. These efforts include various reports by the U.S. National Institute of Standards and Technology [1], the Workshop on the Economics of Information Security (WEIS) [2], and consensus efforts such as those conducted by the Center for Internet Security [3] and the 2004 Corporate Information Security Working Group [4]. However, while they inform and influence one another, information security metrics are not software security metrics.

Consequently, in FY09, CERT began new research in software security measures that builds on CERT's core competence in software and information security. The purpose of this research is to address the following two questions:

- How do I establish and specify the required/desired level of security for a specific software application, set of applications, software-reliant system, system of systems, supply chains, and other multi-system environments?
- How do I measure, at each phase of the development or acquisition life cycle, that the required and/or desired level of security has been achieved?

Approaches to answering the first question define the baseline against which software security can be measured. Such approaches create a meaningful measure of the degree of software security for a specific set of related software components. Ideally, this measurement is performed as part of initial planning and specification, not as an afterthought during testing and integration.

In addition to demonstrating that security requirements are satisfied, risk analysis approaches, including the prioritization of software components based on their contribution to mission success, are also relevant. The SEI has undertaken promising work to identify methods, such as assurance cases, for capturing this expression [5]. This research task will examine the suitability of these methods in establishing a foundation for measuring software security. It will also recommend a range of alternatives with appropriate selection criteria. Software development project managers and stakeholders will be able to select from these alternatives to define a required level of security as part of their software validation criteria.

Given a baseline against which to measure, approaches to the second question will include key product measures, process measures, and performance indicators that can be used to validate the required level of software security appropriate to a given life cycle phase. Such measures will be developed within the context of a measurement process and framework that can be tailored for a specific development project. Table 1 presents early examples of life-cycle-phase measures that could be used to validate required levels of software security:

Research tasks in FY10 include

- investigating existing bodies of knowledge to lay the foundation for addressing the two presented research questions
- building relationships with key thought leaders and potential collaborators
- identifying core definitions
- developing an initial software security measurement process
- publishing initial findings

Research tasks in FY11 and beyond include

- organizing FY10 results by software development life cycle phase to inform the development of a software security measures framework and updated process
- identifying software security measures for acquisition—defining measures that can be written into requests for proposal (RFPs), contracts, service level agreements, and to assist in making funding decisions
- integrating software security development and acquisition measures into selected security assessment and evaluation instruments as well as selected software development and measurement standards

## References

- [1] Chew, Elizabeth, et. al. *Performance Measurement Guide for Information Security: Special Publication 800-55 Revision 1*. National Institute of Standards and Technology (NIST), July 2008. <http://csrc.nist.gov/publications/nistpubs/800-55-Rev1/SP800-55-rev1.pdf>. Additional NIST reports are available at <http://csrc.nist.gov/publications/PubsSPs.html>.
- [2] The Ninth Workshop on the Economics of Information Security (WEIS 2010). <http://weis2010.econinfosec.org/index.html>. Agendas and papers presented from prior WEIS events are available online.
- [3] The Center for Internet Security Consensus Information Security Metrics. <http://cisecurity.org/securitymetrics.html>.
- [4] Corporate Information Security Working Group. Adam H. Putnam, Chairman; Subcommittee on Technology, Information Policy, Intergovernmental Relations & the Census Government Reform Committee, U.S. House of Representatives. "Report of the Best Practices and Metrics Teams." November 17, 2004; updated January 10, 2005. <http://net.educause.edu/ir/library/pdf/CSD3661.pdf>.
- [5] Work in assurance cases at the Software Engineering Institute: <http://www.sei.cmu.edu/dependability/tools/assurancecase/index.cfm>.

Life cycle phase	Example software security measures
Requirements engineering	<ul style="list-style-type: none"> <li>• Percentage of relevant software security principles reflected in requirements specifications (this assumes that security principles essential for a given development project have been selected)</li> <li>• Percentage of security requirements that have been subject to analyses (risk, feasibility, cost/benefit, performance tradeoffs) prior to being included in the specification</li> <li>• Percentage of security requirements covered by attack patterns, misuse/abuse cases, and other specified means of threat modeling and analysis</li> </ul>
Architecture and design	<ul style="list-style-type: none"> <li>• Percentage of architectural/design components subject to attack surface analysis and measurement</li> <li>• Percentage of architectural/design components subject to architectural risk analysis</li> <li>• Percentage of high-value security controls covered by security design patterns</li> </ul>
Coding	<ul style="list-style-type: none"> <li>• Percentage of software components subject to static and dynamic code analysis against known vulnerabilities and weaknesses</li> <li>• Percentage of defects discovered during coding that was injected in architecture and design; in requirements specification</li> <li>• Percentage of software components subject to code integrity and handling procedures, such as chain of custody verification, anti-tampering, and code signing</li> </ul>
Testing	<ul style="list-style-type: none"> <li>• Percentage of defects discovered during testing that was injected in coding; in architecture and design; in requirements specification</li> <li>• Percentage of software components with demonstrated satisfaction of security requirements as represented by a range of testing approaches (functional, risk-based, fuzz, penetration, black box, white box, code coverage, etc.)</li> <li>• Percentage of software components that demonstrated required levels of attack resistance and resilience when subject to attack patterns, misuse/abuse cases, and other specified means of threat modeling and analysis</li> </ul>

*Table 1: Example Software Security Measures by Life Cycle Phase*