Measuring Software Security Assurance

Principal Investigators:
Christopher Alberts, Julia Allen, and Robert Stoddard

Problem Addressed

Many organizations measure just for the sake of measuring, with little or no thought given to what purpose and business objectives are being satisfied or what questions each measure is intended to answer. However, meaningful measurement is about transforming strategic direction, policy, and other forms of management decision into action and measuring the performance of that action.

Effective measures express the extent to which objectives are being met, how well requirements are being satisfied, how well processes and controls are functioning, and the extent to which performance outcomes are being achieved. The basic goal of measurement and analysis is to provide decision makers with the information they need, when they need it, and in the right form. In recent years, researchers have begun to turn their attention to the topic of software security assurance and how to measure it.

Software security assurance is justified confidence that software-reliant systems are adequately planned, acquired, built, and fielded with sufficient security to meet operational needs, even in the presence of attacks, failures, accidents, and unexpected events. For several years, various groups within the software engineering community have been working diligently to identify practices aimed at developing more secure software. However, efforts to measure software security assurance have yet to materialize in any substantive fashion, although some foundational work has been performed [1].

As a result of the software engineering community's interest, the CERT® Program at Carnegie Mellon University's Software Engineering Institute (SEI) has chartered the Security Measurement and Analysis (SMA) Project to advance the state-of-the-practice in security measurement and analysis. The SMA Project builds on the CERT Program's core competence in software and information security as well as the SEI's work in software engineering measurement and analysis. The purpose of this new research project is to address the following three questions:

- How do we establish, specify, and measure justified confidence that a software-reliant product is sufficiently secure to meet operational needs?
- How do we measure at each phase of the development or acquisition life cycle that the required/desired level of security has been achieved?
- How do we scale measurement and analysis approaches to complex environments, such as large-scale, networked, software-reliant systems (e.g., systems of systems)?

In essence, the three research questions examine how decision makers (e.g., development program and project managers as well as acquisition program officers) can measure and monitor the security posture of large-scale, networked, software-reliant systems across the life cycle and supply chain.

Research Approach

Our research approach comprises the following activities:

- survey existing measurement and analysis approaches
- identify any limitations in existing approaches relevant to their application to large-scale, networked systems
- develop a framework for measuring the security characteristics of large-scale, networked systems
- develop a suite of methods and tools for implementing the framework

Our survey of traditional security measurement and analysis approaches indicated that they do not readily scale to today's large-scale, networked, software-reliant systems [1]. As a result, decision makers lack confidence in the security characteristics of their software infrastructures.

Traditional measurement and analysis approaches are based on the principle of system decomposition and component analysis, where the first step is to decompose a system into its constituent components. Next, the individual components are prioritized, and only the most critical components are analyzed in detail. Limitations of traditional approaches include the following:

- Only critical components are analyzed; non-critical components and interdependencies among components are not addressed.
- Causal relationships are presumed to be simple, direct, and linear. Non-linear relationships, such as feedback, are not analyzed.
- Confidence in the performance of critical components is not sufficient for establishing confidence in the performance of the parent system (or the parent system of systems).

Based on our research, we developed the SEI Integrated Measurement and Analysis Framework (IMAF), which is shown in Figure 1. IMAF employs systemic analysis to integrate subjective and objective data from a variety of sources, including targeted analysis, status reporting, and measurement, to provide decision makers with a consolidated view of the performance of large-scale, networked systems.

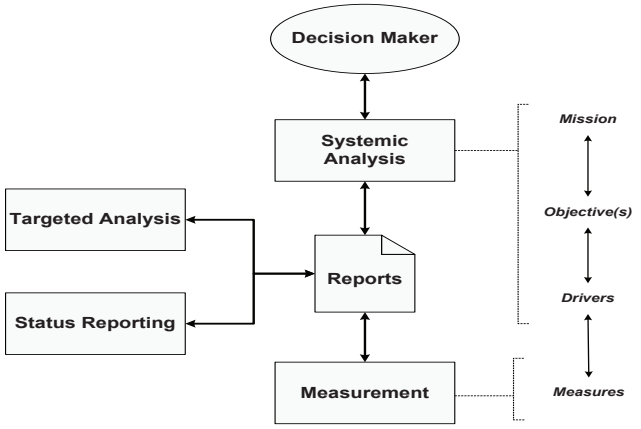

Figure 1: SEI Integrated Measurement and Analysis Framework (IMAF)

Systemic analysis is based on system theory. The underlying goal of system theory is to analyze a system as a whole rather than decomposing it into individual components and then analyzing each component separately [2]. In fact, some properties of a system are best analyzed by considering the entire system, including

- influences of environmental factors
- feedback and nonlinearity among causal factors
- systemic causes of failure (as opposed to proximate causes)
- emergent properties

The SEI approach for conducting systemic analysis requires identifying and analyzing a set of factors that have a strong influence on a system's mission and objectives. These factors are called drivers [3]. Figure 1 shows how drivers enable decision makers to link the security mission and objectives to measures that provide insight into a system's security characteristics. SEI experience shows that effective performance assessment requires approximately 15 to 25 drivers.

To assess secure development of software-reliant systems, we identified a total of 17 drivers. Nine drivers focus on programmatic issues: program security objectives, security plan, contracts, security process, security task execution, security coordination, external interfaces, organizational and external conditions, and event management. The remaining eight drivers examine product and operational attributes: security requirements, security architecture and design, code security, integrated system security, adoption barriers, operational security compliance, operational security preparedness, and product security risk management.

Finally, as illustrated in Figure 2, we have started to develop the following methods for implementing IMAF:

- The Software Security Review (SSR) is a method conducted by independent teams to assess the security characteristics of software-reliant systems. SSR is a driver-based approach that can be used to measure and monitor software security assurance across the life cycle and supply chain (including acquisition, development, and operations).
- Model-Based SSR incorporates predictive analytics, such as Bayesian Belief Networks (BBNs), into its analysis approach. Model-Based SSR enables quantitative analysis of software security assurance using a combination of subjective and objective data.
- Multi-View Decision Making (MVDM) is a coordinated approach for applying multiple security assessment methods. MVDM uses SSR to provide a broad view of software security assurance. An assessment team can use the findings of SSR to select and perform follow-on, "deep-dive" assessments. MVDM helps optimize security assessment activities by applying resources where and when they are most needed.

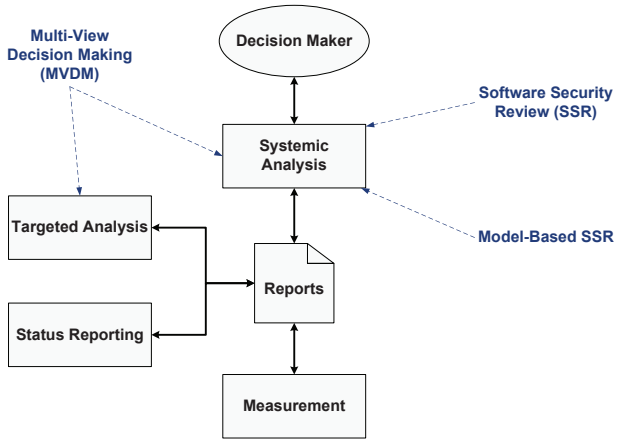

Figure 2: SEI measurement methods for software security assurance

IMAF and its associated methods provide a unique approach for software security measurement and analysis because they

- assess the behavior of large-scale, networked, software-reliant systems as a whole
- enable analysis of complex interrelationships and dependencies among a system's components
- establish justified confidence in the security characteristics of large-scale, networked systems across the life cycle and supply chain

We are currently beginning to pilot IMAF and its associated methods.

Expected Benefits

Expected benefits of this research include the following:

- Decision makers will have better tools for predicting and diagnosing security-related problems and for making well-informed decisions about security.
- IMAF and its associated methods will provide justified confidence in the security of software-reliant products that are acquired, developed, deployed, and sustained by acquisition and development programs.
- IMAF and its associated methods will provide a robust platform for conducting research in any security domain that requires measurement and analysis.

2010 Accomplishments

The 2010 accomplishments of the SMA Project include the following:

- developed the initial version of IMAF
- developed a prototype set of drivers for secure development of software-reliant systems
- initiated development of the SSR and MVDM assessment methods
- identified candidate security practices and measures related to selected drivers from the prototype set
- performed an initial mapping of security standards NIST 800-53 and ISO 27002 to the prototype set of drivers for secure development of software-reliant systems
- developed a notional Bayesian Belief Network using the prototype set of drivers

Future Goals

In 2011, we plan to make progress in the following areas:

- begin piloting the SSR and MVDM methods
- use the results of these pilots to refine IMAF, SSR, and MVDM as appropriate
- use the results of pilots to revise the prototype set of drivers for secure development of software-reliant systems
- begin development of driver sets focused on other parts of the life cycle and supply chain
- continue mapping security standards to driver sets
- develop Bayesian Belief Networks for selected driver sets
- begin development of Model-Based SSR
- mine data from SSR and MVDM pilots to identify a baseline set of software security measures
- explore applying IMAF to other security domains, such as incident management and operational security management

References

[1] Christopher Alberts, Julia Allen, and Robert Stoddard, "Integrated measurement and analysis framework for software security," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, CMU/SEI-2010-TN-025, 2010. *http://www.sei.cmu.edu/library/abstracts/reports/10tn025.cfm*

[2] Nancy Leveson, "A new accident model for engineering safer systems," *Safety Science,* vol. 42, no. 4 pp. 237-270, April 2004.

[3] Christopher Alberts and Audrey Dorofee, "A framework for categorizing key drivers of risk," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, CMU/SEI-2009-TR-007, 2009. *http://www.sei.cmu.edu/library/abstracts/reports/09tr007.cfm*