Measuring the Software Security Requirements Engineering Process

Nancy Mead

July 2012

ABSTRACT: Although there has been much research work in security requirements engineering, we do not have adequate ways of measuring this and other security engineering processes. In this paper, we study a measurement approach to security requirements engineering, align it with the Security Quality Requirements Engineering (SQUARE) method, and use both the original and revised security requirements measurement approach to analyze projects that were developed with and without SQUARE.

INTRODUCTION

In recent years there has been a lot of research in the area of software security requirements engineering [1, 2]. There are now so many distinct approaches that survey papers and reports have been developed to compare and contrast the various methods [3]. However, in the course of performing our security requirements engineering research, we have for the most part been unable to produce the measurement data needed to demonstrate that using these methods will actually lead to improved software security.

We are just now starting to see measurement models that provide the needed support for measuring software security aspects across the life cycle [4]. Although we will see that many of these process measurement approaches are subjective, they represent a start toward obtaining real data to show that approaches to security requirements engineering will result in improved security in the as-built system.

In this paper, we study a measurement approach to security requirements engineering, align it with the Security Quality Requirements Engineering (SQUARE) method, and use both the original and revised security requirements measurement approach to analyze projects that were developed with and without SQUARE.

First, we discuss the software security measurement and analysis activity at the Software Engineering Institute (SEI) [4], focusing on the driver considerations for security requirements. Next we briefly describe the SQUARE methodology, which has been well documented and discussed in depth elsewhere [5, 6, 7, 8].

Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

Phone: 412-268-5800
Toll-free: 1-888-201-4479

www.sei.cmu.edu

With the SQUARE methodology in mind, we examine and revise the security requirements driver considerations. Next we apply the original and revised considerations to an actual project developed using SQUARE. We further apply the revised considerations to a project that was already in development when specific SQUARE steps were incorporated into the process. We conclude with a discussion of the utility of this measurement approach and future work needed for both the measurement and requirements engineering aspects.

II. SOFTWARE SECURITY MEASUREMENT AND ANALYSIS

The goal-question-metric paradigm [9] has long focused on having a reason for measuring, rather than just doing measurement and collecting data for its own sake. The software security measurement and analysis (SSMA) project (http://www.cert.org/sse/measurement.html) focuses on measurement for the objective of addressing goals such as formulating a business case or demonstrating improved software quality, with questions formulated to support those objectives and measurement data, in turn, providing needed support.

Once there is an understanding of the utility of the measurement data, meaningful data can be collected and used to effect change and to measure improvement. Recently, research projects such as SSMA have begun to turn their attention to the topic of software security assurance and how to measure it.

The Committee on National Security Systems defines software assurance as follows [10]:

> *Software assurance (SwA) is the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in the intended manner.*

A related definition from the SSMA project for software security assurance is [4]:

> *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.*

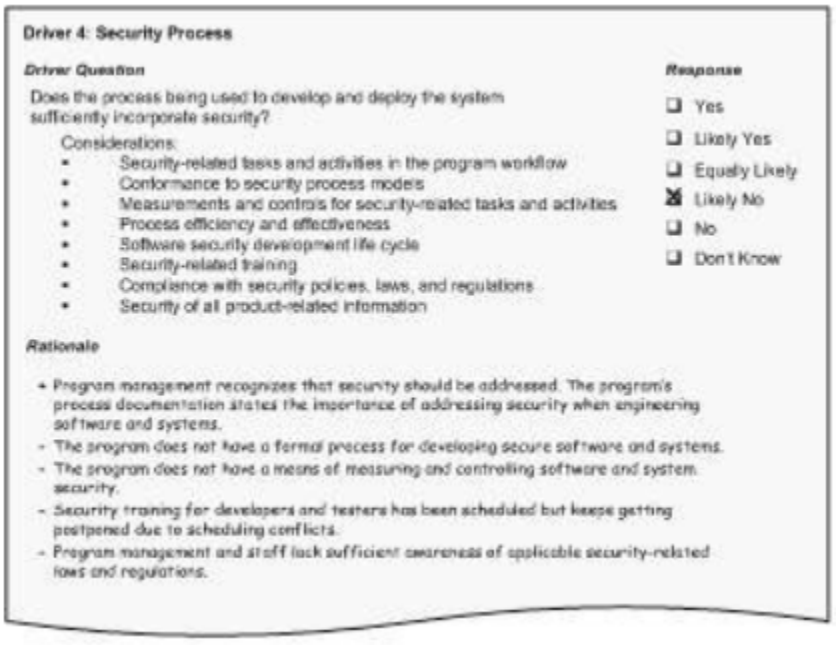
The purpose of the SSMA work is to address the following two questions:

> How do we establish, specify, and measure justified confidence that interactively complex software-reliant systems are 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?

Driver identification establishes a set of factors (drivers) that are used to measure a program's performance relative to its mission and objectives. Each driver identified should have a strong influence on whether objectives are achieved. A prototype set of 17 drivers for software security has been identified by the SSMA project. These are Program Security Objectives, Security Plan, Contracts, Security Process, Security Task Execution, Security Coordination, External Interfaces, Organizational and External Conditions, Event Management, Security Requirements, Security Architecture and Design, Code Security, Integrated System Security, Adoption Barriers, Operational Security Compliance, Operational Security Preparedness, and Product Security Risk Management.

Each driver has an associated driver question, and for each driver a set of considerations has been identified to help answer the driver question. An example question, response, and rationale are given for Driver 4, Security Process, in Fig. 1:



Driver 4: Security Process

Driver Question

Does the process being used to develop and deploy the system sufficiently incorporate security?

Considerations:

- Security-related tasks and activities in the program workflow
- Conformance to security process models
- Measurements and controls for security-related tasks and activities
- Process efficiency and effectiveness
- Software security development life cycle
- Security-related training
- Compliance with security policies, laws, and regulations
- Security of all product-related information

Response

- ❑ Yes
- ❑ Likely Yes
- ❑ Equally Likely
- ☒ Likely No
- ❑ No
- ❑ Don't Know

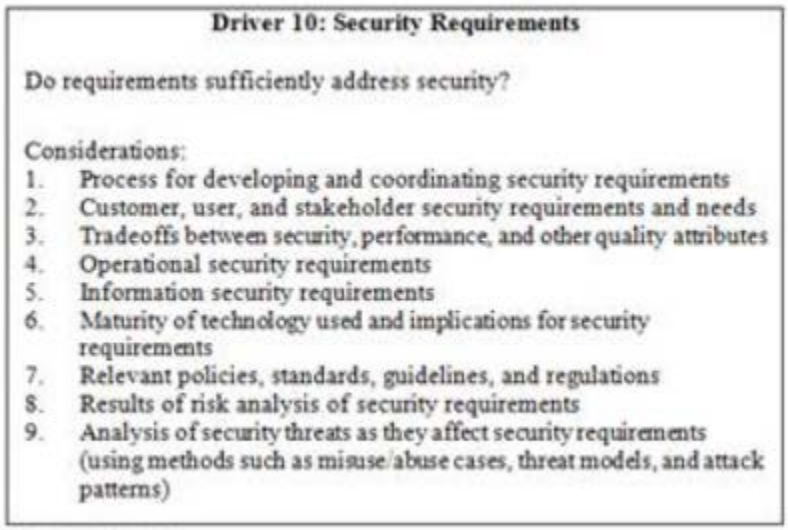
Rationale

- + Program management recognizes that security should be addressed. The program's process documentation states the importance of addressing security when engineering software and systems.
- - The program does not have a formal process for developing secure software and systems.
- - The program does not have a means of measuring and controlling software and system security.
- - Security training for developers and testers has been scheduled but keeps getting postponed due to scheduling conflicts.
- - Program management and staff lack sufficient awareness of applicable security-related laws and regulations.

Figure 1: Question, Response, and Rationale for Driver 4, Security Process

The focus of this paper will be on Driver 10: Security Requirements, with the associated question, "Do requirements sufficiently address security?" The draft considerations are shown in Table I.

Table I: Draft Considerations



Driver 10: Security Requirements

Do requirements sufficiently address security?

Considerations:

1. Process for developing and coordinating security requirements
2. Customer, user, and stakeholder security requirements and needs
3. Tradeoffs between security, performance, and other quality attributes
4. Operational security requirements
5. Information security requirements
6. Maturity of technology used and implications for security requirements
7. Relevant policies, standards, guidelines, and regulations
8. Results of risk analysis of security requirements
9. Analysis of security threats as they affect security requirements (using methods such as misuse/abuse cases, threat models, and attack patterns)

We will first address the question of whether the considerations map to an existing security requirements engineering process and then provide an example response and rationale analogous to the one given for Driver 4 in Fig. 1.

III. SECURITY REQUIREMENTS ENGINEERING WITH SQUARE

When security requirements are considered at all during the system development life cycle, they tend to be general lists of security features such as password protection, firewalls, virus detection tools, and the like. These are, in fact, not security requirements at all but rather implementation mechanisms that are intended to satisfy unstated requirements, such as authenticated access. As a result, security requirements that are specific to the system and that provide protection of essential services and assets are often neglected. In reviewing requirements documents, we typically find that security requirements, when they exist, are in a section by themselves and have been copied from a generic set of security requirements. The requirements elicitation and analysis needed to get a better set of security requirements seldom takes place.

Researchers in the CERT® Program at the SEI have developed a methodology to help organizations build security into the early stages of the production life cy-

---

® CERT is a registered mark of Carnegie Mellon University.

cle. SQUARE consists of nine steps that generate a final deliverable of categorized and prioritized security requirements. Although SQUARE could likely be generalized to any large-scale design project, it was designed for use with information technology systems.

SQUARE involves the interaction of a team of requirements engineers and the stakeholders of an IT project. The requirements engineering team can be thought of as external consultants, though often the team is composed of one or more internal developers of the project. When SQUARE is applied, the user of the method should expect to have identified, documented, and inspected relevant security requirements for the system or software that is being developed.

SQUARE begins with the requirements engineering team and project stakeholders agreeing on technical definitions that serve as a baseline for all future communication. Next, assets are identified, and business and security goals are outlined. Third, artifacts and documentation are created, which are necessary for a full understanding of the relevant system. A structured risk assessment determines the likelihood and impact of possible threats to the system. Following this work, the requirements engineering team determines the best method for eliciting initial security requirements from stakeholders. The choice of method depends on several factors, including the stakeholders involved, the expertise of the requirements engineering team, and the size and complexity of the project. Once a method has been established, the participants rely on artifacts and risk assessment results to elicit an initial set of security requirements. Two subsequent stages involve categorizing and prioritizing these requirements for management's use in making trade-off decisions. Finally, an inspection stage ensures the consistency and accuracy of the security requirements that have been generated. This process is depicted in Fig 2.

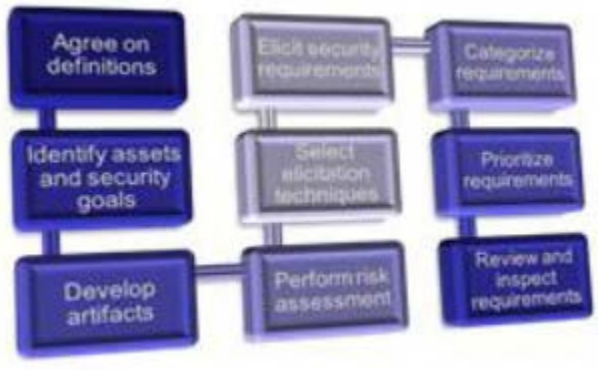


Figure 2: The SQUARE Process

IV. A REVISED SET OF CONSIDERATIONS FOR SECURITY REQUIREMENTS

The driver considerations in Table I were examined side by side with the SQUARE process steps to identify the differences and develop a revised set of considerations. Consideration 1 examines the existence of a security requirements process (such as SQUARE) and, obviously, is needed. Consideration 2 could be revised to reflect the participation of requirements engineers in the process as well as stakeholders. In the SQUARE process, stakeholders include users and customers. Consideration 3, while important, takes place outside the development of security requirements, so it would not be included. However, SQUARE steps 7 and 8, categorize and prioritize security requirements, could be substituted for this consideration. Consideration 4, operational security requirements, is not emphasized in the SQUARE process, but it is an important class of security requirements, so it remains. Consideration 5, information security requirements, would be the output from SQUARE and corresponds to SQUARE step 6. Consideration 6 remains valid, although in SQUARE it probably would be part of step 4, perform risk assessment. Consideration 7 would not come out of the SQUARE process but would be among the artifacts that support SQUARE in step 3. Consideration 8 corresponds to SQUARE step 4. Consideration 9 is valid, although in SQUARE it would be subsumed under steps 3 and 4. There are no corresponding considerations for SQUARE steps 1 and 2; these should be included. SQUARE step 5, select elicitation techniques, would be covered under consideration 1, which deals with process, as would SQUARE step 9.

A revised and reordered set of considerations is shown in Table II.

Table II: Revised Set of Considerations

	Considerations
1.	Process for developing and coordinating security requirements *(Existence of a process such as SQUARE)*
2.	Agree on definitions and identify assets and security goals *(SQUARE steps 1 and 2)*
3.	Relevant policies, standards, guidelines, and regulations *(SQUARE step 3)*
4.	Results of risk analysis of security requirements *(SQUARE step 4)*
5.	Maturity of technology used and implications for security requirements *(SQUARE step 4)*
6.	Analysis of security threats as they affect security requirements (using methods such as misuse/abuse cases, threat models, and attack patterns) *(SQUARE steps 3 or 4)*
7.	Customer, user, and stakeholder security requirements and needs, developed in conjunction with requirements engineers *(SQUARE steps 5 and 6)*
8.	Information security requirements *(SQUARE step 6)*
9.	Security requirements categorization and prioritization *(SQUARE steps 7 and 8)*
10.	Operational security requirements *(Should appear in SQUARE step 6)*

V. Assessment of Security Requirements Considerations Based on Actual Projects

The original and revised considerations were then applied to an actual project as a proof of concept. First we will look at one of the case studies where SQUARE was applied [11]. A brief description of the project follows:

The Acme Company is a private company headquartered in Pittsburgh with a staff of approximately 1,000 across multiple offices in the United States. It provides technical and management services to various public sectors and a number of diversified private companies.

ABC Services is one of four major subsidiaries of the Acme Company. ABC provides a range of specialized services for asset management. With over 15 years of experience, ABC developed the Asset Management System (AMS). This software product provides a tool for companies to make strategic allocations and planning of their critical IT assets. AMS is an Executive Asset Management Information System that provides decision support capabilities via customized views. These views are displayed in graphical forms and consist of information such as asset information, operational performance, and other user-defined metrics.

AMS also integrates with many third-party software suites to provide enterprise-level services and features. Archibus/FM, which is used internally, is a facility

infrastructure management and operation tool that supports all aspects of infrastructure management. It is also fully integrated with AutoCAD, an industry standard software application that ensures proper change management. All changes made on architectural drawings are immediately reflected in the database. Another integrated tool is a backend Geographical Information System (GIS) used to organize information and geographic locations by sites.

Overall, the AMS Software Suite is a full-service support product in all aspects of infrastructure management and facility-related services.

Using the original considerations, we might have the evaluation and rationale found in Table III.

Table III: Application of Original Considerations to an Actual Project

Driver 10: Security Requirements

Do requirements sufficiently address security?

Considerations:

1. Process for developing and coordinating security requirements
2. Customer, user, and stakeholder security requirements and needs
3. Tradeoffs between security, performance, and other quality attributes
4. Operational security requirements
5. Information security requirements
6. Maturity of technology used and implications for security requirements
7. Relevant policies, standards, guidelines, and regulations
8. Results of risk analysis of security requirements
9. Analysis of security threats as they affect security requirements (using methods such as misuse/abuse cases, threat models, and attack patterns)

Rationale:

+ The project has a process for developing and coordinating security requirements that involves all stakeholder groups.
+ The project develops information security requirements, considering risk and threat analysis.
+ The project considers the technology used and relevant artifacts such as policies, standards, etc.
- The process does not incorporate tradeoffs with other quality attributes, but it does prioritize security requirements.
- The process does not consider operational security requirements except peripherally.

Response: Likely Yes

Now let's consider how the same project would fare under the revised considerations, as shown in Table IV.

Table IV: Application of Revised Considerations to an Actual Project

Driver 10: Security Requirements

Do requirements sufficiently address security?

Considerations:

1. Process for developing and coordinating security requirements *(Existence of a process such as SQUARE)*
2. Agree on definitions and identify assets and security goals *(SQUARE steps 1 and 2)*
3. Relevant policies, standards, guidelines, and regulations *(SQUARE step 3)*
4. Results of risk analysis of security requirements *(SQUARE step 4)*
5. Maturity of technology used and implications for security requirements *(SQUARE step 4)*
6. Analysis of security threats as they affect security requirements (using methods such as misuse/abuse cases, threat models, and attack patterns) *(SQUARE steps 3 or 4)*
7. Customer, user, and stakeholder security requirements and needs, developed in conjunction with requirements engineers *(SQUARE steps 5 and 6)*
8. Information security requirements *(SQUARE step 6)*
9. Security requirements categorization and prioritization *(SQUARE steps 7 and 8)*
10. Operational security requirements *(Should appear in SQUARE step 6)*

Rationale:

+ The project has a process for developing and coordinating security requirements that involves all stakeholder groups and requirements engineers.
+ The project considers definitions, assets, and security goals.
+ The project develops information security requirements, considering risk and threat analysis.
+ The project considers the technology used and relevant artifacts such as policies, standards, etc.
- The process does not consider operational security requirements except peripherally.

Response: Yes or Likely yes, depending on whether operational security requirements are considered.

In the overall scheme of things, the worst case would be a project that did not have a requirements engineering process at all. If we considered a project that did not have a security requirements process, it would fail to satisfy many of the considerations for both the original and revised Driver 10. For example, a project that had a requirements engineering process that was not specific to security would run into difficulties with revised driver considerations 2, 4, 5, 6, 8, 9, and 10.

We will consider one small example here, a project that we worked with after development had started [12]. This project can be briefly described as follows:

VAD Corporation is a privately held, medium-sized commercial organization. The VADSoft project is a financial application. Application functionality is determined by the internal client and user roles and functions. User roles and functions are defined by the VAD Corporation's security model.

The project did not have documented security goals, so we elicited security goals from the project team. We found that they had conducted a risk assessment but had not considered security risks. Their risk assessment focused primarily on management and schedule risks. To facilitate the VADSoft team in eliciting and prioritizing risks, the SQUARE team provided them with a list of risks that the project may face based on their requirements documents and inputs from meetings. The VADSoft team then could update this list with other risks that they anticipated. A risk matrix was also prepared and provided by the SQUARE team to the VADSoft team to help them assess the risk exposure of the project.

If we had not intervened, VADSoft would have had the results shown in Table V. Even with our intervention, the results would have been spotty, as the project did not have the time to review or incorporate into the software the security requirements identified on their behalf. The results in the table are for only the revised considerations, but they would be similar under the original considerations.

Table V: Application of Revised Considerations to a Project without Security Goals

Driver 10: Security Requirements

Do requirements sufficiently address security?

Considerations:

1. Process for developing and coordinating security requirements *(Existence of a process such as SQUARE)*
2. Agree on definitions and identify assets and security goals *(SQUARE steps 1 and 2)*
3. Relevant policies, standards, guidelines, and regulations *(SQUARE step 3)*
4. Results of risk analysis of security requirements *(SQUARE step 4)*
5. Maturity of technology used and implications for security requirements *(SQUARE step 4)*
6. Analysis of security threats as they affect security requirements (using methods such as misuse/abuse cases, threat models, and attack patterns) *(SQUARE steps 3 or 4)*
7. Customer, user, and stakeholder security requirements and needs, developed in conjunction with requirements engineers *(SQUARE steps 5 and 6)*
8. Information security requirements *(SQUARE step 6)*
9. Security requirements categorization and prioritization *(SQUARE steps 7 and 8)*
10. Operational security requirements *(Should appear in SQUARE step 6)*

Rationale:

- The project does not have a process for developing and coordinating security requirements.
- The project did not consider definitions, assets, and security goals.
- The project did not do risk analysis for security requirements.
- The project develops information security requirements but does not do risk or threat analysis.
- The project did not consider the technology used or relevant artifacts such as policies, standards, etc.
- \+ The project considers operational security requirements.
- Security requirements are not categorized or prioritized.

Response: No or Likely no, depending on credit given for the attempt to include security.

Unfortunately, many projects still use a “one size fits all” requirements process that focuses primarily on end-user functionality and fails to adequately consider security. A project that has a requirements process that specifically addresses security, regardless of the details of the process, is likely to fare the best against the considerations for Driver 10.

VI. CONCLUSIONS AND FUTURE PROCESS MEASUREMENT WORK

We started with an objective to provide a mechanism for measuring security requirements engineering process. By merging the results of the software security measurement and analysis activity with the SQUARE process, we were able to assess the security requirements engineering process for two actual projects.

Although the results are interesting and certainly an improvement over the state of the practice in measuring the security requirements engineering process, much more work is needed. First, on the requirements side, we need to examine other security requirements engineering processes to see if further considerations should be added to the security requirements driver. On the measurement side, we need to apply the original and revised security driver considerations to projects that were developed using a variety of security requirements approaches or none at all. We also need to go through a similar exercise with all the drivers and their considerations and apply them to real project(s). In addition, we need to recognize that processes such as SQUARE may not go far enough in identifying operational security requirements and addressing activities, such as tradeoff analysis, that take place after the relevant security requirements have been identified.

We also need to review the considerations to determine whether they are sufficiently objective. As of right now, they are very much focused on the process rather than the product and depend on the expertise of the assessor.

REFERENCES

[1] Inger Anne Tøndel,Martin Gilje Jaatun, Per Håkon Meland, "Security requirements for the rest of us: A survey," *IEEE Software,* January 2008, pp. 20-27.

[2] Jeffrey A. Ingalsbe, Louis Kunimatsu, Tim Baeten, Nancy R. Mead, "Threat modeling: Diving into the deep end," *IEEE Software,* January 2008, pp. 28-34.

[3] M. U. A. Khan and M. Zulkernine, "On selecting appropriate development processes and requirements engineering methods for secure software," in *Proc. 33rd Annual IEEE International Computer Software and Applications Conference*, Seattle, WA, 2009, pp. 353-358.

[4] C. Alberts, J. Allen, R. Stoddard, "Risk-based measurement and analysis: Application to software security," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA,

Technical note CMU/SEI-2011-TN-032, 2011.

[5] N. R. Mead, E. D. Hough, and T. R. Stehney II, "Security quality requirements (SQUARE) methodology," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Technical report CMU/SEI-2005-TR-009, 2005.

[6] N. R. Mead and T. Stehney, "Security quality requirements engineering (SQUARE) methodology," Presented at *27th Software Engineering for Secure Systems (SESS05 at ICSE 2005),* St. Louis, M0, SESS05 retrieved from

http://homes.dico.unimi.it/%7Emonga/ sess05.html

[7] N. R. Mead, V. Viswanathan, and D. Padmanabhan, "Incorporating security requirements engineering into the dynamic systems development method," *Proc. International Workshop on Security and Software Engineering* (at International Computer Software and Applications Conference), Turku, Finland, 2008.

[8] J. Caulkins, E. D. Hough, N. R. Mead, and H. Osman, "Optimizing investments in security countermeasures: A practical tool for fixed budgets," *IEEE Security & Privacy,* vol. 5, no. 5, pp.24–27.

[9] V. Basili, "Applying the Goal/Question/Metric Paradigm in the Experience Factory," in *Software Quality Assurance and Measurement: Worldwide Perspective,* N. Fenton, R. Whitty, and Y. Lizuka, eds., London, U.K: International Thomson Computer Press, 1995, pp. 21-44.

[10] Committee on National Security Systems (CNSS). Instruction No. 4009, National Information Assurance Glossary. Revised June 2009.

[11] P. Chen, N. R. Mead, M. Dean, L. Lopez, D. Ojoko-Adams, H. Osman, and N. Xie, "SQUARE Methodology: Case study on asset management system," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Special report CMU/SEI-2004-SR-015, 2004.

[12] A. Gayash, V. Viswanathan, D. Padmanabhan, N. R. Mead, "SQUARE-Lite: Case study on VADSoft project," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Special report CMU/SEI-2008-SR-017, 2008.

[13] Shari Lawrence and Rachel Rue Pfleeger, “Cybersecurity economic issues: Clearing the path to good practice,” *IEEE Software,* January 2008, pp. 35-42.

This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

DM-0001120