Towards Measurement of Security Effectiveness Enabling Factors in Software Intensive Systems

Reiho M. Savola

Abstract—Adequate information security effectiveness during system operation is the ultimate goal of all security solutions for software-intensive systems. Sufficient and credible measurement of security effectiveness supports informed decision-making in engineering and management practices throughout the system development life cycle. Although detail-level security metrics can be developed for special purposes, their meaningfulness suffers if their relationship to the overall security effectiveness objectives cannot be traced. This paper analyzes the factors contributing to security effectiveness of software-intensive systems.

Index Terms—Security, security metrics, security effectiveness, software-intensive systems.

I. INTRODUCTION

As Software-Intensive Systems (SISs) incorporate increasingly more critical applications and ever more complex and networked software and devices, they become exposed to information security risks in increasing numbers. The classical perimeter-oriented approaches to security are becoming outdated; the perimeter is growing non-existent or at least fuzzy. A networked environment creates potential for most faults to become security vulnerabilities. The increased exposure of software-intensive systems has coincided with attackers’ increased awareness of the multitude of vulnerabilities present in software [1].

The core challenge is to be able to develop and deploy effective security solutions for operational systems in a cost effective and efficient way regardless of the complexity and openness of the systems. Information security effectiveness is a major challenge during system operation, and it must be supported adequately by the choices made in development. Consequently, sufficient system quality should be supported throughout the System Development Life Cycle (SDLC), and the software and security engineers developing the systems and the operations personnel who handle their administration should be able to understand factors contributing to security effectiveness in a uniform manner. Currently, security work is done mostly in silos: developers do not think enough about the actual operation situation, while operations personnel do not give feedback to the developers, or their perception of technical security solutions is biased, or even ‘black and white’. Information security system management often tends to consider technical solutions in a simplistic perimeter-oriented way: ‘Just add an anti-virus system and firewalls, and increase the length of passwords.’ End users in organizations and in general are not security professionals and cannot be expected to take on too much of the burden of achieving security effectiveness. Certainly, they cannot be highly aware of even major security risks and vulnerabilities associated with the system they are using. Security work is a complex socio-technical challenge and is not effective when approached from only one or a few perspectives. Development-time penetration testing and focusing on information security management processes are good partial solutions [2] but are not enough for security effectiveness.

Recently, security metrics have received growing interest from researchers and practitioners alike. Many of the traditional efforts at security measurement have been unsatisfactory. [3]. The poor applicability of earlier efforts is, to a great extent, due to the lack of a holistic and widely accepted model for reasoning about security effectiveness.

The main contribution of this study is identification and analysis of available evidence contributing to security effectiveness of software-intensive systems that can be applied in security measurement practices.

Section II discusses the background of this study: security risks, and sine related measurements and metrics. Section III overviews goals of security measurement, and Section IV analyses the factors affecting security effectiveness and their measurability in view of the literature and state-of-the-art knowledge. Section V gives a rundown of related work and a comparison of our work with that most closely related to ours. Finally, our conclusions and perspectives for the future are presented in Section VI.

II. BACKGROUND

A Software-Intensive System is a system in which software plays a fundamental role, influencing the design, construction, deployment, and evolution of the whole system [4]. Software includes elements such as programs, procedures, data, and instructions for system control and information processing [5]. In the course of the SDLC, software must be designed, implemented, configured, maintained, and disposed of. SISs are used in applications of various business domains, among them telecommunications, industrial automation, manufacturing, logistics, and information and service systems of organizations. Typically, systems of these kinds are complex, networked, and highly distributed. An SIS or parts of it are often tailored and/or configured to be used for several applications and domains. In addition to the complexity challenge, a considerable difficulty from the perspective arises from the fact...
that not all applications of the System under Investigation (SuI) are known by its developers during the design and implementation phases. When component-based software engineering is applied, a software component may be used in several applications in a variety of systems in the course of its service life. ‘ICT system diffusion’ – the trend of deploying Information and Communication Technology (ICT) systems in domains such as industry automation – is part of this problem, with systems that were developed for other kinds of applications. SISs can be developed via many, quite different processes, e.g. agile and lean techniques or spiral methods.

To enable reasoning about the nature of security effectiveness, below we first discuss the nature of information security risks. Table I summarizes the key terminology and definitions used in this study.

<table>
<thead>
<tr>
<th>Term</th>
<th>Ref.</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security effectiveness</td>
<td>Savola [7], Jansen [8], ITSEC [9]</td>
<td>Assurance that the stated security objectives are met in the SuI and the expectations for resiliency in the use environment are satisfied, while the SuI does not behave other than intended.</td>
</tr>
<tr>
<td>Security correctness</td>
<td>Savola [7], Jansen [8], ITSEC [9]</td>
<td>Assurance that the security controls have been correctly deployed in the SuI. In practice, the aim is to measure how close the security controls deployed in a real-world system are in line with their ideal deployment as defined by the security objectives.</td>
</tr>
<tr>
<td>Security efficiency</td>
<td>Savola [7]</td>
<td>Assurance that adequate security quality has been achieved in the SuI, meeting the resource, time, and cost constraints.</td>
</tr>
<tr>
<td>Security objective</td>
<td>ISO/IEC 15408 [10]</td>
<td>A high-level statement of intent to counter identified threats and/or satisfy identified organisational security policies and/or assumptions.</td>
</tr>
<tr>
<td>Security requirement</td>
<td>ISO/IEC 15408 [10]</td>
<td>A requirement, stated in standardised language, that is meant to contribute to reaching of the security objectives.</td>
</tr>
</tbody>
</table>

Note that the US Committee on National Security Systems [13] defines software assurance as ‘the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its lifecycle and that the software functions in the intended manner’. This definition focuses on software vulnerabilities. From the security effectiveness perspective, the term ‘security assurance’, used in this study, is more appropriate, the focus of it being on the security performance of the SuI in its operation environment. According to McGraw [2], software security addresses software-induced security risks and how to manage them. Although this definition is more limited in scope, the scope of McGraw’s approach [2] to software security partly overlaps the scope of this study.

There would not be any need for security solutions without security risks. For simplicity, we focus here on information security risks. The connection between a security risk and the severity of its impact is important for security-related and business decision-making. For one to be able to manage the security risk of SISs, it is of utmost importance to understand the business context. The business impact of a risk is expressed by cost (e.g., loss of revenue, reputation, trust, opportunities, employee morale, or safety; increased exposure to lawsuits; or harm to environmental assets). In simple terms, there cannot be significant risk without potential for significant loss [14]. The most common method used in assessing the impact of security risks is to use an oversimplified risk probability x severity of impact matrix. This kind of risk assessment addresses the epistemological view, measuring the human judgements about security risk. Relating the technical risks of an SIS to business impact is often very challenging.

Software vulnerabilities originate from design-level flaws and implementation bugs, both manifested as exploitable faults, which, in turn, represent vulnerabilities. According to McGraw [2], half of all vulnerabilities originate from the design and half from the implementation. The more faults are exploitable, the more possibilities there are for the attackers to use them as entry points or otherwise compromise the system, increasing the attack surface. Attacks targeting software faults have become very sophisticated. They often exploit unexpected sequences of multiple, often non-contiguous faults (referred to as Byzantine faults) throughout the software [1].

The notion of security risk in software has entered common discourse, yet researchers and software developers have only recently begun systematically studying how to build secure software [2]. Sometimes, confusion is caused because the ‘risk’ in the information security field is often treated as something predictable, persistent, and measurable [15]. One of the core security problems in current SISs is that growth in software size seems to increase also the number of vulnerabilities [2]: ‘the more lines of code, the more vulnerabilities’. On the other hand, source-code analysis tools rely on the assumption that ‘fewer faults in source code mean the software is less vulnerable.’ [1].

Note that traditional software risk management is not purely carried out for security, encompassing also reliability and safety [2]. Security threats to an SIS are intentional. However, unintentional faults in the system can be exploited also intentionally, consequently turning into security threats. In contrast, ‘pure’ reliability and safety threats are unintentional. Vulnerabilities, flaws or weaknesses in a system that could be exploited, are often emphasized more in security Risk Management (RM) than in general RM.
III. SECURITY MEASUREMENT, METRICS AND THEIR OBJECTIVES

It is often claimed that an activity cannot be managed well if it cannot be measured. The term ‘security metrics’ refers to metrics depicting the information security level or performance of the SUI – a technical system, product, service, or organisation [7]. Measurement is a process of experimentally obtaining information about the magnitude of a quantity [16]. A measurement result indicates single-point-in-time data for this quantity, while metrics are descriptions of data derived from measurements used to facilitate decision-making [7]. Standardisation (e.g., the ISO/IEC 27004 standard [17]) has adopted the term “measure” in the same meaning as “metric”.

In practice, current software and system measurement instruments do not address the full breadth of assurance and security as a measure of quality [18]. Security metrics approaches are still in their childhood. Measuring security is a challenging task. In fact, security of SISs cannot be measured as a universal concept, on account of the systems’ complexity, uncertainty, non-static nature, limited observability of operational systems; and attackers’ malice [19], [20]. Therefore, terms such as ‘indicators’ or ‘strength’ might be more appropriate in the case of security measurement objectives. However, in this study, ‘measurement’ is used in a broad sense, encompassing also assessment wherein more accurate measurement is impossible. Similarly, the term ‘metrics’ is used for indicators. One of the mistakes often made in security measurement is to focus too much on the metrics themselves (the ‘trees’) at detail level [3] and not the actual evidence and its quality (the ‘forest’). This is at least partly due to the lack of meaningful abstractions to support decomposition of the security measurement challenge into sub-challenges. The dependencies of the sub-challenges should be traced. Fig. 1 provides an example of high-level decomposition of authentication objectives created for security metrics [21].

Although security cannot be measured as a whole, it still can be measured via suitable reference requirements, which may be based on, for example, security risk analysis results, best practice and regulations. To some extent, the security level and performance can be increased by means of best practice, but the lack of more precise risk perception implies a greater number of unknown and illusory risks, and it could result in costly and incorrect security solutions.

The users of security metrics include system developers, security engineers, and information security managers. Moreover, certain security metrics can be used in automatic monitoring and adaptive control of SISs. Examples of more in-depth needs of various stakeholders in system development and operation are listed in [22].

According to [7], there are three fundamental objectives of security measurement: 1) security correctness, 2) security effectiveness, and 3) security efficiency. Examples of security correctness measurements are risk management, security objective compliance, requirement compliance, and compliance with regulations [23]. The above-mentioned security measurement objectives are abstractions that are helpful in striving for meaningful security metrics and associated measurements. Intuitively, security correctness relies heavily on high-quality RM and adequacy of security objectives. Risk-driven proactive security effectiveness cannot be built on something that is not sufficiently correct from a security perspective. Security correctness is a necessary but not inherently sufficient
requirement for adequate security effectiveness. Moreover, in practice, a balanced trade-off is needed between effectiveness and efficiency. Security efficiency measurement has a sideline role from a security effectiveness angle. Complete security effectiveness cannot be achieved in realistic systems even when one resorts to state-of-the-art techniques, because of the inherent uncertainty of security risks.

The security effectiveness of SISs suffers from the fact that software developers focus on satisfying correctness criteria but often fail to meet effectiveness criteria, particularly under extraordinary unanticipated conditions [8].

IV. FACTORS CONTRIBUTING TO SECURITY EFFECTIVENESS

We now discuss and analyse selected types of evidence contributing to security effectiveness in SISs, in light of the literature, knowledge of the state of the art, and our earlier work in security metrics. In the analysis, we do not address security efficiency, because it can be seen as a horizontal dimension to security effectiveness, requiring a balanced trade-off with the latter. This analysis investigates the following factors that contribute to security effectiveness: evidence of 1) direct security effectiveness, 2) quality of risk assessment, 3) security correctness, and 4) system quality. The selection of the factors is based on the areas of emphasis found in the literature and our previous work [7].

It is important to understand the business contexts wherein the SIS is used if one wishes to develop effectiveness metrics and measurements. Here, we discuss security effectiveness from two perspectives: that of the SIS itself from a security standpoint, and the organizational operation context in which the SIS is used as a part of an information system controlled by the organization’s Information Security Management (ISM) practices. Most work on security effectiveness has focused on the latter context, while, few research results are available that address security effectiveness as experienced by end users elsewhere than in organizations. Moreover, Blakley et al. [24] assert that very little is known about the effectiveness of security measures in general, the main reason being that there are many incentives to suppress information about the incidents that occur. The factors affecting security effectiveness as perceived by such end users are probably quite similar to the factors in organizations but scale differently.

Even though there is a paucity of discussions of security effectiveness, most contributions in software, network, computer, and IT security use the term ‘security’ or ‘security performance’ synonymously with ‘security effectiveness’, which refers to the obvious ultimate goal. Moreover, factors such as software quality, vulnerability management and security correctness are discussed as if direct contributors to ‘security’. The lack of effectiveness thinking is also visible from the fact that the role of risk assessment is not properly understood or even referred to in many contributions, not to mention software engineering practices.

A. Direct Evidence of Security Effectiveness

From a technical system perspective, usually the security effectiveness of the SuI can be measured with high accuracy only over (typically quite long) spans of the actual operation of the system, when it is exposed to potential for materialisation of real security risks.

Measuring security effectiveness directly is not straightforward, and in practice this is possible only in part. The ideal way of doing it would be to gather evidence of the complete collection of risks that the security controls are intended to counter. For example, the evidence may include log data from critical security mechanisms and frequent penetration tests. This would be a case of direct evidence of the strength of the security controls deployed. The only problem is that the actual frequency of occurrence of those risks is not static, so there are no data on which one could base solid estimate. Risks to the systems may not materialise because no attacks are actually taking place or because only impotent ones are encountered by the system—in other words luck is with the system-owner.

Penetration testing is used often to obtain evidence of security effectiveness in the latter phases of system development. It is especially valuable when carried out in the actual operation environment. However, as real-world payloads seldom can be used during the development phases, it is difficult to determine the direct effect of penetration testing on the security effectiveness of the SuI. Regardless, the testing environment should be as close as possible to the actual operation situation of the SuI, whether it is a part of an organization’s information system or used by a consumer end user. Many SISs are used in both kinds of applications and domains. Consequently, Research and Development (R&D) for SISs needs to implement sufficient support for successful information security management in both of these contexts. Often, the latter context is completely lacking in information security management practices, or the practices are only rudimentary.

B. Evidence of Risk Assessment and Its Quality

Security cannot be managed well without proper RM. Some approaches to RM emphasize vulnerabilities (e.g., NIST SP 800-30 [25] and Microsoft Risk Management Framework (RMF) [26]), while others apply a more general concept of risk. Vulnerabilities and risk do not have a one-to-one mapping. RMFs for managing organizational risk have been used for some years now. More recently, the framework approach has been applied to security risks’ management and, even more recently, specifically to software risk management in software [1].

A sub-activity of RM, Risk Assessment (RA) refers to the identification, analysis, and evaluation of risks. RA deals with multiple levels of uncertainty, including ‘known unknowns’ [27], [28] (known but undemonstrated risks) and even ‘unknown unknowns’ [27], [28] (risks that are not known to exist). The goal of RM is to reduce and control the uncertainty of the ‘unknowns’. Because of the inherent uncertainty, RA cannot generate anything more accurate than security risk perception. This fact is important to understand since we need to base security objectives on the risk perception offered by RA and the RM decisions, regulations, and/or best practice.

The difference between actual and perceived security risk can be studied through investigation of ontological and epistemological views of it, where the former represents the
actual real world and the latter of it as perceived by humans. Oscarson [29] discusses three distinct cases at ontological and epistemological level: 1) known risks, 2) unknown risks, and 3) delusional risks. At epistemological level, there is a model of known risks that corresponds to its counterpart at ontological level; however, unknown things are present only at ontological level, and delusional elements are present only at epistemological level. High-quality RA increases the predictability of unknown things and minimises the bias caused by delusional risks. One source of delusional risk perception is blind reliance on security best practice, which is not necessarily applicable to the situation in question.

C. Evidence of Security Correctness

Security correctness is a key factor in the level of effectiveness a security control can reach. Ideally, all security is manageable via security controls. Up-to-date security controls are designed to mitigate security risks in the application scenarios. RA results and RM decisions are reflected in the security objectives, which are further processed into security requirements. The latter, in turn, become part of the system and unit requirements in practical systems. This ideal control structure is a typical assumption in current security standards, such as the ISO/IEC 27000 [6] family of information security management standards, and the technical system certification standard Common Criteria (CC) ISO/IEC 15408 International Standard [10].

In SISs, security correctness has to take into account the operation environments where the SuI will be used, including regulatory and security requirements. In this ideal model, measurement of security correctness can be carried out by comparison of the security controls described in security objectives and regulations to the actual system. Security effectiveness, on the other hand, is concerned instead with the strength of the security controls, their ability to withstand attacks launched within or beyond the organizational (or SuI) boundaries, or mitigation of the effect of such attacks in the event of their success.

D. Evidence of Security-Related System Quality

Unfortunately, the ideal security control model is not realistic for practical secure software engineering. Hayden [3] claims that the security control concept is myopic, and McGraw [2] adds that a software security problem is more likely to arise because trouble with a system’s standard issue portion than in a mechanism implemented on the basis of security objectives. In fact, the principles and requirements on which measurement of correctness is based purport to guarantee the attainability of a given set of objectives related to security. Accordingly, it is unanimously agreed that, for a firewall to protect a system against, for instance, external attacks, it should not be configured to allow any incoming connections. It is, therefore, hard to discuss effectiveness in isolation from correctness, although the existence of security correctness does not always guarantee security effectiveness; what will be the case only, if the policy against which the correctness is measured is airtight against vulnerabilities. In short, correctness implies effectiveness only if the reference for the correctness measurement itself is ‘correct’ in terms of its quality. It seems clear that one cannot list everything in the security requirements in practice, and a great many code-level issues affect software security [2]. Design specifications often miss important security details that appear only in code [30]. Therefore, system correctness and security correctness can be seen as separate measurement objectives with many interdependencies.

In fact, there are two correctness paths in striving for security effectiveness in SISs: 1) to minimise security problems arising from the SuI on account of issues with system quality, and 2) to maximise the effect of security controls. The scope of system quality is wide. Therefore, in security work, the focus is on system quality as a lens for improving security effectiveness.

V. RELATED WORK

To the best of our knowledge, no previous integrative studies of security effectiveness of software-intensive systems have been made available. However, security effectiveness has been addressed in standardisation.

Regardless of several major attempts to standardise security evaluation of technical systems, they have achieved only limited success in advancing security’s measurability [8]. This is largely because the standards are rigid and created for certification and carrying out these processes requires significant amounts of time and money. Not enough resources are available and the focus is often on meeting requirements rather than actually being secure. The most widely spread of these efforts is the CC (ISO/IEC 15408) Standard [10], which focuses primarily on documentation rather than the actual security effectiveness of the system in operation. There is no meaningful basis for assurance cases that can be used to verify security as a property of software as opposed to security correctness [1].

The ISO/IEC 27004 standard [17] addresses measurement, reporting, and improvement in the effectiveness of IS Management Systems (ISMS) as defined in ISO/IEC 27001. According to this standard, the effectiveness assessment in organizations addresses policies, security risk management, objectives, controls, processes and procedures. ‘Base measure’ and ‘derived measure’ abstractions are used. Detailed guidelines for selection of measurement targets are not given.

VI. CONCLUSIONS AND FUTURE WORK

This study has identified the following factors as contributor to the holistic perception of security effectiveness in software-intensive systems: evidence of direct security effectiveness, quality of risk assessment, security correctness, and system quality.

Analysis underscores that, more thorough modelling and pilot studies are needed for definition of a rigorous methodology enabling systematic development of security effectiveness metrics. Our future work includes industrial experimentation and iterative development of heuristics for security effectiveness metrics and measurements.

REFERENCES

Reijo Savola received the degree of master of science in electrical engineering from the University of Oulu, 1992, and the degree of Licentiate of Technology in computer science from the Tampere University of Technology, 1995.

He is currently working as a principal scientist at VTT Technical Research Centre of Finland in Oulu, Finland. Previously he has worked as a software engineer for Elektrobit Group Plc. in Oulu, Finland and in Redmond, WA, United States. His current research interests include security metrics and security requirements engineering. He has experience in information security, software engineering, telecommunications, multi-technology engineering topics, and digital signal processing algorithms.

Mr. Savola is an author of 109 journal, conference, and workshop papers and has been chair, program chair, and member of technical committees in several conferences and workshops. He is acting as the Editor-in-Chief of the International Journal on Advances in Security.