IDENTIFICATION OF BASIC MEASURABLE SECURITY COMPONENTS IN SOFTWARE INTENSIVE SYSTEMS

Reijo M. Savola

VTT Technical Research Centre of Finland

P.O. Box 1100, 90571 Oulu, Finland

ABSTRACT

Appropriate information security solutions for software-intensive systems, together with evidence of their security performance help to prevent serious consequences for businesses and the stakeholders. Security metrics can be used to offer this evidence. We investigate practical and holistic development of security metrics for software-intensive systems. Our approach is security requirement-centric. The high-level security requirements are expressed in terms of lower-level measurable components applying a decomposition approach. Detailed security metrics are developed based on the basic measurable components identified at the leaf level of the decomposition.

KEY WORDS

Security metrics, security measurements, security requirements

IDENTIFICATION OF BASIC MEASURABLE SECURITY COMPONENTS IN SOFTWARE INTENSIVE SYSTEMS

1 INTRODUCTION

The increasing complexity of software-intensive and telecommunication products, together with pressure from security and privacy legislation, are increasing the need for adequately validated security solutions. To obtain evidence of the information security performance of systems needed for the validation, services or products, systematic approaches to measuring security are needed. The field of defining security metrics systematically is very young. Because the current practice of security is still a highly diverse field, holistic and widely accepted measurement and metrics approaches are still missing.

The rest of this paper is organized in the following way. Section 2 gives a short introduction to security metrics. Section 3 introduces the proposed security metrics development process. Section 4 discusses threat and vulnerability analysis, and the next section security requirements. Section 6 describes decomposition of security requirements. Section 7 explains issues important in the measurement architecture and evidence collection, Section 8 discusses the further steps of metrics development. Section 9 presents related work and finally, Section 10 summarizes the study with some future research questions and conclusions.

2 SECURITY METRICS

Security metrics and measurements can be used for decision support, especially in assessment and prediction. When using metrics for prediction, mathematical models and algorithms are applied to the collection of measured data (e.g. regression analysis) to predict the security performance. The target of security measurement can be, e.g., an organization, its processes and resources, or a product or its subsystem. In general, there are two main categories of security metrics: (i) security metrics based on threats but not emphasizing attacker behavior, and (ii) security metrics predicting and emphasizing attacker behavior. In this

study, we concentrate in the former type of metrics. Security metrics properties can be quantitative or qualitative, objective or subjective, static or dynamic, absolute or relative, or direct or indirect. According to ISO 9126 standard [1], a direct measure is a measure of an attribute that does not depend upon a measure of any other attribute. On the other hand, an indirect measure is derived from measures of one or more other attributes. See [2] and [3] for examples of security metrics.

3 PROPOSED SECURITY METRICS DEVELOPMENT PROCESS

In this study, we use the following iterative process for security metrics development, partly based on [4]. The steps for the process are as follows:

1. Carry out threat and vulnerability analysis. Identify and elaborate threats of the system under investigation and its use environment. If enough information is available, identify known or suspected vulnerabilities. This work can continue iteratively as more details of the target will be known.
2. Define and prioritize security requirements, including related requirements critical from security point of view, in a holistic way based on the threat and vulnerability analysis. The most critical security requirements should be paid the most attention. Pay attention to the simplicity and unambiguity of the requirements.
3. Identify *Basic Measurable Components* (BMC) from the higher-level security requirements using a decomposition approach. BMCs relate the metrics to be developed to security requirements.
4. Develop measurement architecture for on-line metrics and evidence collection mechanisms for off-line metrics.
5. Select BMCs to be used as the basis for detailed metrics based on their feasibility and criticality.
6. Define and validate detailed security metrics, and the functionalities and processes where they are used.

The steps are iterative and the order of the steps can be varied. Steps 1 and 2 should be started as early as possible in the research and development lifecycle and elaborated iteratively as the system design becomes more mature. If possible, steps 3 and 4 can be carried out in parallel to each other. Step 4 can be initiated already during the architectural design phase provided that suitable information is available.

4 THREAT AND VULNERABILITY ANALYSIS

Threat analysis is the process of determining the relevant threats to an SUI (System under Investigation). The outcome of the threat analysis process is preferably a prioritized description of the threat situations. In practice, there are many ways to carry out threat analysis, from simply enumerating threats to modeling them in a more rigorous way. The extent of threat analysis depends, e.g., on the criticality of the use cases in the SUI. The following threat and vulnerability analysis process can be used, based on the Microsoft threat risk modeling process [5]: (1) identify security objectives, (2) survey the SUI architecture, (3) decompose the SUI architecture to identify functions and entities with impact to security, (4) identify threats, and (5) identify vulnerabilities.

The security objectives can be decomposed, e.g., to identity, financial, reputation, privacy and regulatory and availability categories [6]. There are many different sources of risk guidance that can be used in developing the security objectives, such as laws, regulations, standards, legal agreements and information security policies. Once the security objectives have been defined, it is important to analyze the designed SUI architecture and to identify different components, data flows and trust boundaries. To identify the functions and entities with impact to security objectives, the architecture can be decomposed further. Threats are the goals of the adversary and for a threat to exist it must have a target asset. To identify threats, the following questions can be asked [7]:

1. How can the adversary use or manipulate the asset to modify or control the system, retrieve or manipulate information within the system, cause the system to fail or become unusable, or gain additional rights?
2. Can the adversary access the asset without being audited, or skip any access control checks, or appear to be another user?

The threats can be classified using a suitable model like STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) [5]. DREAD (Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability) [5] is a classification scheme for quantifying, comparing and prioritizing the amount of risk presented by each evaluated threat.

Vulnerability analysis can be carried out after appropriate technological choices have been made. Vulnerabilities in the technology and implementation affect to threats of the system. In vulnerability analysis, well-known vulnerability listings and repositories such as OWASP (Open Web Application Security Project) Top 10 [6] can be used. Metrics from Common Vulnerability Scoring System (CVSS) [8] can be used to depict how easy or hard it is to access and exploit a known vulnerability in the system.

5 SECURITY CRITICAL REQUIREMENTS

Security requirements derive from *threats*, *policies* and *environment properties*. Security requirements that are derived from threats are actually countermeasures. Security policies are security relevant directives, objectives and design choices that are seen necessary for the system under investigation. Environment properties contribute to the security of the SUI from outside – either advancing or reducing it. The explanation for the security-advancing effect of the environment is that it could to contain a countermeasure solution against a threat, outside the SUI. In general, every security risk due to a threat chosen to be cancelled or mitigated must have a countermeasure in the collection of security requirements. In general, the state of practice in defining security requirements is not at matured level. According to [9], the most current software requirement specifications are either (i) totally silent regarding security, (ii) merely specify vague security goals, or (iii) specify commonly used security mechanisms (e.g., encryption and firewalls) as architectural constraints. In the first case security is not taken into account in an adequately early phase of design. In the second case vague security goals (like “the application shall be secure”) are not testable requirements. The third case may unnecessarily tie architectural decisions too early, resulting in an inappropriate security mechanism. Security requirements are often conceived solely as non-functional requirements along with such aspects as performance and reliability within the requirements engineering community [10]. From the security engineering viewpoint this is a too simplified way of thinking; security cannot be represented only by non-functional requirements since security goals often motivate new functionality, such as monitoring, intrusion detection and access control, which, in turn, need functional requirements. Unfortunately, satisfactory

approaches to capturing and analyzing non-functional requirements have yet to mature [11].

6 DECOMPOSING REQUIREMENTS

The core activity in the proposed security metrics development process is the decomposition the security requirements. In the following, we discuss the decomposition process and give an example of it.

6.1 Decomposition Process

The following decomposition process (based on [12]) is used to identify measurable components from the security requirements:

1. Identify successive components from each security requirement (goal) *that are essential and contribute to the success* of the goal.
2. Examine the subordinate nodes to see if further decomposition is needed. If so, repeat the process with the subordinate nodes as current goals, breaking them down to their essential components.
3. Terminate the decomposition process when none of the leaf nodes can be decomposed any further, or further analysis of these components is no longer necessary. When the decomposition terminates, all leaf nodes should be measurable components.

6.2 Example Decomposition: Authentication

In general, the model depicted in Fig. 1 can be used for high-level authentication decomposition [12] during the process of identifying potential metrics for authentication performance. Similar decompositions can be defined for authorization, confidentiality, integrity, availability and so on.

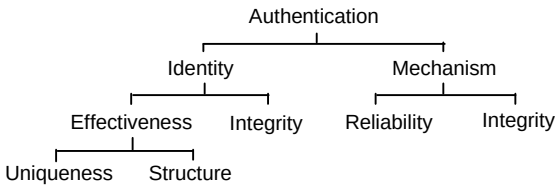

Figure 1. Decomposition of authentication

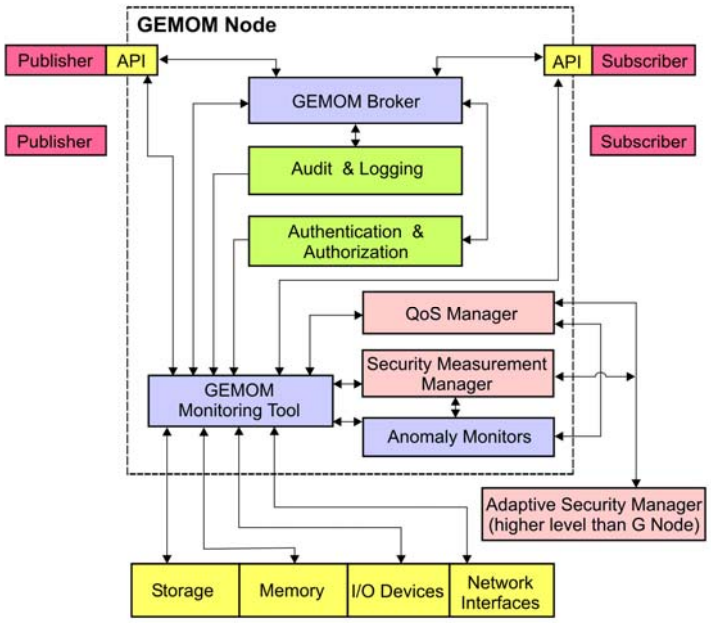

Figure 2. Example of information flows.

Different authentication mechanisms (e.g. password authentication and various forms of biometrics and any combination) can be used for different authentication needs. Fig. 1 commends that the security level of authentication mechanisms is depending on their level of reliability and integrity. There are many ways to use metrics and their combinations.

7 MEASUREMENT ARCHITECTURE AND EVIDENCE COLLECTION

In the case of on-line metrics, the measurement architecture and data flow needs to be designed, in parallel to the overall architectural and data flow design of the SUI. Similarly, in the case of off-line metrics, the evidence collection mechanisms and criteria need to be planned. In many cases, on-line and off-line measurements can be dependent on each other.

Identification of measuring points and development of evidence collection mechanisms can be carried out, e.g., from data flow diagrams and protocol descriptions. As an example, Fig. 2 shows a conceptual picture of information flows of a distributed messaging system (GEMOM,

Genetic Message Oriented Secure Middleware [13]). Security metrics of the Security Monitor module can use information from the Broker module, Audit and Logging module and the Authentication and Authorization module. In addition, the metrics get information from memory, storage, Input-Output devices and network interfaces.

8 DETAILED METRICS DEVELOPMENT

The detailed development of chosen security metrics includes formalizing the metric to a computational form. Different weights can be associated to different metrics to indicate the relative importance or weights among the components. A “close to correct” weight assignment is critical, since in practice there are no analytical results for determining the relative priorities of the elements besides careful use of one’s expertise and judgment [12].

8.1 Authentication and Authorization Metrics

Use of authentication mechanisms from different authentication categories makes the authentication stronger, the categories being: (i) something you know, (ii) something you have and (iii) something you are. *Authentication strength value* (e.g. from 0 to 1) can be assigned. In the case of multi-modal authentication the security strength value can be increased. In a similar way, strengths can be assigned to different authentication mechanisms, algorithms and protocols. In addition to metrics that measure the performance of authentication, metrics that express the attacker behaviour can be developed, such as (i) number of authentication failures, (ii) proportion of failed authentications, and (iii) a measure of authentication trends. False positives in authentication are attackers falsely permitted access and false negatives are authorized users who are hindered from accessing the systems they should be able to use. Regarding federated identity management and single sign-on, typical use patterns based on use cases can be defined or recorded from the system. The actual patterns from logs can be compared to the typical use patterns [4].

Most authorization metrics can be based on the log and metadata information of the users and objects they access or trying to access. This data can be used to investigate authorization mechanism use trends and to track extraordinary user behaviour. In addition, metrics from CVSS

(Common Vulnerability Scoring System) [8] can be used to illustrate how easy or hard it is to access and exploit a known vulnerability in the system. Leaving a known vulnerability in the system might be a deliberate choice decided in the risk management process. CVSS's *access vector metric* measures whether the vulnerability is exploitable locally or remotely and *access complexity metric* measures the complexity of attack required to exploit the vulnerability once an attacker has access to the target system or service [4].

8.2 Confidentiality and Integrity Metrics

Cryptographic confidentiality strength metrics measure the performance of cryptographic protection used to ensure the end-to-end confidentiality of messages, logs and metadata. Different algorithms can be used based on the level of confidentiality needed. The protection in physical media (storage and memory) and the protection from unauthorized access to them is important. The reliability and effectiveness of access control are important too. *Confidentiality impact metric* of CVSS measures the impact on confidentiality of a successful exploit of vulnerability in the system. As in the case of confidentiality, cryptographic integrity strength metrics measure the level of cryptographic protection used to ensure the data integrity in messages, metadata, logs and storages (persistent data). *Integrity impact metric* of CVSS measures the impact to integrity of a successfully exploited vulnerability (none, partial, complete) [4].

8.3 Availability and Non Repudiation Metrics

Availability metrics from safety and reliability engineering can be used to measure the availability dimension. *Availability impact metric* of CVSS measures the impact to availability of a successfully exploited vulnerability (none, partial, complete).

In non-repudiation, it is important that proof-of-identity evidence can be obtained from the system. The evidence should be consistent, reliable and its integrity should be protected. Consistency, reliability and integrity metrics can be used for non-repudiation. Cryptographic strength metrics can be used to measure the performance of cryptographic algorithms used to ensure the non-repudiation of messages [4].

8.4 Metrics based on Other Requirements

Some other requirements potentially have effect to the security performance of the system. Application-level and business requirements should be taken into account in the security metrics development. Note that business environment and constraints affect a lot the impact and exposure of security risks. Usability and performance of security solutions are very important design objectives [4].

8.5 Doubts about Security Metrics

The feasibility of measuring security and developing security metrics to present actual security phenomena has been criticized in many contributions. In designing a security metric, one has to be conscious of the fact that the metric simplifies a complex socio-technical situation down to numbers or partial orders. McHugh [15] is skeptical of the side effects of such simplification and the lack of scientific proof. Bellovin [16] remarks that defining metrics is hard, if not infeasible, because an attacker's effort is often linear, even in cases where exponential security work is needed. Another source of challenges is that luck plays a major role [17] especially in the weakest links of information security solutions. Those pursuing the development of a security metrics program should think of themselves as pioneers and be prepared to adjust strategies as experience dictates [14].

9 RELATED WORK

Wang and Wulf [12] describe a general-level framework for measuring system security based on a decomposition approach. CVSS [8] (Common Vulnerability Scoring System) is a global initiative designed to provide an open and standardized method for rating information technology vulnerabilities from a practical point of view. NIST's Software Assurance Metrics and Tool Evaluation (SAMATE) project [18] seeks to help answer various questions on software assurance, tools and metrics. OWASP (Open Web Application Security Project) [6] contains an active discussion forum on security metrics. More security metrics approaches are surveyed in [2] and [3].

10 CONCLUSIONS AND FUTURE WORK

Feasible and widely accepted approaches for security metrics development of software-intensive systems are still missing. We have introduced a novel methodology for security metrics development based on threats, policies, security requirements and requirement decomposition. The methodology is highly iterative and the order of steps can be varied depending on the information available.

Further work is needed in the development of generic and application and domain specific security requirement model decompositions, ways to define measurement architectures, evidence collection and selection of measurable components. Furthermore, heuristics for assessment of the feasibility of candidate component metrics are needed. The approach and parts of it need to be validated by experimentation in practical use scenarios originating from different application domains.

11 REFERENCES

[1] ISO/IEC 9126-4, Software Engineering – Product Quality – Part 4: Quality in Use Metrics, 2000.

[2] Savola, R., A Novel Security Metrics Taxonomy for R&D Organisations, Proceedings of the 7th Annual Information Security South Africa (ISSA) Conference, July 7-9, 2008, Johannesburg, South Africa.

[3] Herrmann, D. S., Complete Guide to Security and Privacy Metrics, Auerbach Publications, 2007, 824 p.

[4] Savola, R. and Abie, H., Identification of Basic Measurable Security Components for a Distributed Messaging System. In SECURWARE 2009, June 18-23, 2009, Athens, Greece, 8 p.

[5] Howard, M. and LeBlanc, D., Writing Secure Code, Second Edition, Microsoft Press, 2003.

[6] OWASP (Open Web Application Security Project), Threat Risk Modeling, web reference: www.owasp.org

[7] Swiderski, F. and Snyder, W., Threat Modeling, Microsoft Press, 2004.

[8] Schiffman, M., A Complete Guide to the Common Vulnerability Scoring System (CVSS). White paper.

[9] Firesmith, D., Specifying Reusable Security Requirements, Journal of Object Technology, Vol. 3, No. 1, Jan/Feb 2004, 61-75.

[10] Chung, L., Nixon, B. A., and Yu, E., Using Quality Requirements to Systematically Develop Quality Software, 4th Int. Conference on Software Quality, McLean, VA, October 1994.

[11] Nuseibeh, B. and Easterbrook, S., Requirements Engineering: A Roadmap, The Future of Software Engineering, Special Volume published in conjunction with ICSE 2000, A. Finkelstein, Ed., pp. 35-46.

[12] Wang, C. and Wulf, W. A., Towards a Framework for Security Measurement, 20th National Information Systems Security Conference, Baltimore, MD, Oct. 1997, pp. 522-533.

[13] Abie, H., Dattani, I., Novkovic, M., Bigham, J., Topham, S. and Savola, R., GEMOM – Significant and Measurable Progress Beyond the State of the Art, ICSNC 2008, Sliema, Malta, Oct. 26-31, 2008, pp. 191-196.

[14] Payne, S. C., A Guide to Security Metrics, SANS Institute Information Security Reading Room, 2006.

[15] McHugh, J., Quantitative Measures of Assurance: Prophecy, Process or Pipedream? Workshop on Information Security System Scoring and Ranking (WISSSR), ACSA and MITRE, Williamsburg, VA, May 2001 (2002).

[16] Bellovin, S. M., On the Brittleness of Software and the Infeasibility of Security Metrics, IEEE Security & Privacy, July/August 2006, p. 96.

[17] Burris, P. and King, C., A Few Good Security Metrics, METAGroup, Inc. Oct 2000.

[18] Plack, P. E., SAMATE's Contribution to Information Assurance, IANewsletter, Vol. 9, No. 2, 2006.