Journal of Cybersecurity and Privacy

MDPI

Article

Developing Security Assurance Metrics to Support Quantitative Security Assurance Evaluation

Shao-Fang Wen *, Ankur Shukla and Basel Katt

Department of Information Security and Communication Technology, Norwegian University of Science and Technology, 2815 Gjøvik, Norway

* Correspondence: shao-fang.wen@ntnu.no

Abstract: Security assurance (SA) is a technique that helps organizations to appraise the trust and confidence that a system can be operated correctly and securely. To foster effective SA, there must be systematic techniques to reflect the fact that the system meets its security requirements and, at the same time, is resilient against security vulnerabilities and failures. Quantitative SA evaluation applies computational and mathematical techniques for deriving a set of SA metrics to express the assurance level that a system reaches. Such metrics are intended to quantify the strength and weaknesses of the system that can be used to support improved decision making and strategic planning initiatives. Utilizing metrics to capture and evaluate a system's security posture has gained attention in recent years. However, scarce work has described how to combine SA evaluation while taking into account both SA metrics modeling and analysis. This paper aims to develop a novel approach for the modeling, calculation, and analysis of SA metrics that could ultimately enhance quantitative SA evaluation.

Keywords: security assurance; quantitative approach; security metrics; analytics

Citation: Wen, S.-F.; Shukla, A.; Katt, B. Developing Security Assurance Metrics to Support Quantitative Security Assurance Evaluation. *J. Cybersecur. Priv.* **2022**, *2*, 587–605. https://doi.org/10.3390/jcp2030030

Academic Editor: Danda B. Rawat

Received: 3 June 2022
Accepted: 8 August 2022
Published: 10 August 2022

1. Introduction

With the ubiquity and importance of information and communication technology (ICT) systems nowadays, organizations always have a primary concern that there may be vulnerabilities existing in their working environments that can compromise organizational data, disrupt business services, and jeopardize trust. It is therefore important for organizations to have shreds of evidence that show the security mechanisms are correctly and effectively put in place and carry out their intended functions to prevent, detect, or divert a risk or to reduce its impact on a system's assets [1]. Security assurance (SA) is a technique that helps organizations to appraise the trust and confidence that a system can be operated correctly and securely [2]. In detail, SA evaluates, reports, and monitors the security posture of ICT systems to see whether the security features, practices, procedures, and architecture accurately mediate and enforce the security policy before being disseminated or delivered to the target audience [3]. SA, however, is a complicated concept with many different functions such as technical countermeasures, organizational policies, security procedures, etc. Therefore, measuring the level of confidence is a non-trivial exercise in SA, and making reasonable decisions and prioritizations about the pipelined security tasks is ever more so.

To foster effective SA, there must be systematic techniques to reflect the fact that the system meets its security requirements and, at the same time, is resilient against security vulnerabilities and failures [2]. Quantitative SA evaluation applies computational and mathematical techniques for deriving a set of SA metrics (hereinafter "metrics") to express the assurance level that a system reaches [4]. Researchers have identified advantages of quantitative methods in SA evaluation, including (1) providing models with useful information about the behavior of ICT systems in different contexts, (2) expressing the security with less complicated and more coherent mechanisms, and (3) supporting decision

making by using and comparing metrics [5]. Utilizing metrics to capture and evaluate the security posture of ICT systems has gained attention in recent years. Such metrics are intended to deliberate the assurance aspect of the system security to reliably transfer information [6,7]. A key factor in the success of quantitative SA evaluation is, therefore, the development of appropriate metrics that can consequently provide meaningful information used to answer essential questions. For example, to what extent does the system fulfill the requirements of security standards? How is the presence of vulnerabilities detected, and what is the priority to address first? By analyzing the answers, security analysts and stakeholders can examine the security-related issues and, consequently, identify what areas require improvements and find a way to organize the resources efficiently.

To facilitate proper decision making in such scenarios, there is a need for more methodological methods in advancing toward developing metrics to support quantitative SA evaluation. Despite existing work that is underway, scarce work has described how to combine SA evaluation while taking into account both metrics modeling and analysis. The aim of this paper, therefore, is to complement this research gap by proposing systematic approaches in quantitative SA as well as metric development, including the following components: (1) a quantitative SA metamodel for describing the structure of metrics calculation, (2) a comprehensive set of metrics and the corresponding computation algorisms, and (3) illustrated SA analytics for presenting and interpreting metrics. The rest of this paper is organized as follows. In Section 2, we provide an overview of related work. Our methodological approach for quantitative SA evaluation is introduced in Section 3. Section 4 presents the illustrative security assurance analytics with the proposed metric. Lastly, the conclusion is presented in Section 5.

2. Related Work

Research on security assurance and evaluation methods is vast. In the past, various frameworks and standards have been developed for evaluating security. One of the most representative works is Common Criteria (CC) [8]. The CC is an international standard (ISO/IEC 15408) for the security evaluation of IT products. It provides a set of guidelines and specifications that can facilitate the specification of security functional requirements and security assurance requirements. With the strict, standardized, and repeatable methodology, the CC assures implementation, evaluation, and operation of a security product at a level that is commensurate with the operational environments. Despite being a standard, the drawback of such a comprehensive methodology is that the documentation is complicated and needs a large effort on preparation for the evaluation of a product or service against a specific CC assurance level [9,10]. Some other examples of security maturity models are the Building Security In Maturity Model (BSIMM) [11] and OWASP Software Assurance Maturity Model (OpenSAMM) [12] and OWASP Application Security Verification Standard (ASVS) [13], which are provided for the software security domain. BSIMM is a study of how different organizations deal with software security, which resulted in a software security framework that is organized in 116 activities and 12 practices. Like BSIMM, OpenSAMM is an open software security framework developed by OWASP [14], which provides guidelines on which software security practices should be used and how to assess them. Such maturity models provide frameworks, especially in a qualitative fashion, to evaluate the security posture of the process and culture practiced in an organization. OWASP ASVS provides guidelines for web application security testing and corresponding security controls. It also lists a set of security assurance requirements and an associated qualitative evaluation scheme that consists of three maturity levels.

In the past, however, few efforts have been made to provide a generic approach to quantify the security posture to support security assurance evaluation systematically. Several papers in this research area are highlighted below.

Liu and Jin [15] conducted a study to analyze the security threats and attacks on the WLAN network architecture and developed a security assessment and enhancement system. This system is divided into two subsystems, a security assessment system and a

security enhancement system. The security assessment system is based on fuzzy logic and analyzes the vulnerability of the physical layer (PHY) and medium access control (MAC) layer, key management layer, and identity authentication layer. This approach provides a quantitative value of the security level based on security indexes, whereas the security enhancement system is an integrated, trusted WLAN framework based on the trusted network connection that helps to improve the security level of WLAN. Agrawal et al. [16] used the fuzzy analytical hierarchy process (Fuzzy-AHP) methodology to evaluate usable security. They also assessed the impact of security on usability and the impact of usability on security using a quantitative approach. Katt and Prasher [2] proposed a general-purpose security assurance framework and its assurance evaluation process. The basic components of the proposed framework included are the security assurance scheme, security assurance target, security assurance metrics, security assurance technique, evaluation evidence, and assurance level. The framework and process depend on quantitative security assurance metrics that were developed too. They discussed the advantages of quantitative security assurance metrics considering both the security requirements and vulnerabilities.

Furthermore, several researchers have been working on SA metrics development and calculation. For instance, Pham and Riguidel [17] introduced an aggregational method that can be applied in the calculation of the security assurance value of the whole system when combining several entities, which have been evaluated independently. The effects of the emergent relations are taken into account in the calculation of the security assurance value of an attribute in the context of a system. Ouedraogo et al. [18] take advantage of quantitative risk measurement methodologies to develop metrics for IT infrastructure security assurance evaluation along with aggregation techniques, i.e., the assurance level of a system is a specific combination of assurance levels from underlying components. The main algorithms used for the operational aggregation include the recursive minimum algorithm, the recursive maximum algorithm, and the recursive weighted sum algorithm. Moreover, to help businesses address service security assurance, Ouedraogo [19] presents a set of metrics that can estimate the level of confidence for both consumers and providers. The defined metrics can be categorized into three main areas: security-related metrics (existence, correctness, etc.), security verification-related metrics (coverage of verification, depth of verification, etc.), and privacy-related metrics (data confidentiality and service consumer anonymity).

Some SA metrics methodologies use evidence and arguments over security measure adequacy in a security case to build an acceptable level of confidence in system security. For instance, Rodes et al. [20] propose the use of security arguments by facilitating security metrics that need to be complete and valid and propose a framework for argument assessment that generates and interprets security metrics on the example of software systems. Within the framework, security is quantified in terms of a level of belief, i.e., a confidence level of arguments. Several approaches take advantage of patterns to assess and evaluate system security. In this area, for instance, Heyman et al. [21] associate security metrics with patterns, and exploit the relationships between security patterns and security objectives to enable the interpretation of measurements. Fernandez et al. [22] evaluate the security of architecture by considering different misuse patterns. They propose to analyze how many misuse patterns for architecture can be countered when adding security patterns to improve the architecture. The calculated value then represents the level of security for the applied security patterns. Lastly, Villagrán-Velasco et al. [23] evaluate system security based on threat enumeration and on verifying if these threats are or are not controlled in a specific software architecture. They also consider the effect of policies and the use of weights according to their impact.

3. Our Methodological Approach

Our methodological approach is divided into two parts: a modeling approach for quantitative SA evaluation, and the proposed metrics as well as the corresponding calculation rules based on the SA evaluation model.

3.1. Security Assurance Evaluation Model

Given a complex IT system, direct measurement of its assurance level is generally not possible. Thus, the overall goal of the SA modeling is to transform SA evaluation into measurable works. In this respect, our modeling approach follows a five-level hierarchy, in which each node represents a distinct assurance component, as shown in Figure 1. In our model, security assurance components constitute the essential parts of assurance metrics calculation. The assurance target is the product or system that is the subject under security assessment, such as an information system, part of a system or product, or a cloud ecosystem. The evaluation serves to validate claims made about the assurance target. One core principle behind our proposal is that the confidence in the system security is quantified through two critical assurance perspectives: the protection side and the weakness side of the assurance target. Each perspective is composed of one or more criteria, and each criterion is composed of one or several elements until reaching the lowest level (i.e., assurance conditions). Thereafter, the overall score of an assurance target is estimated from the test results of the assurance conditions and the criteria/elements of the evaluation model applied. These estimates are aggregated continuously in conjunction with predefined algorithms to arrive at a more fine-tuned final estimate at the top-level assurance target.

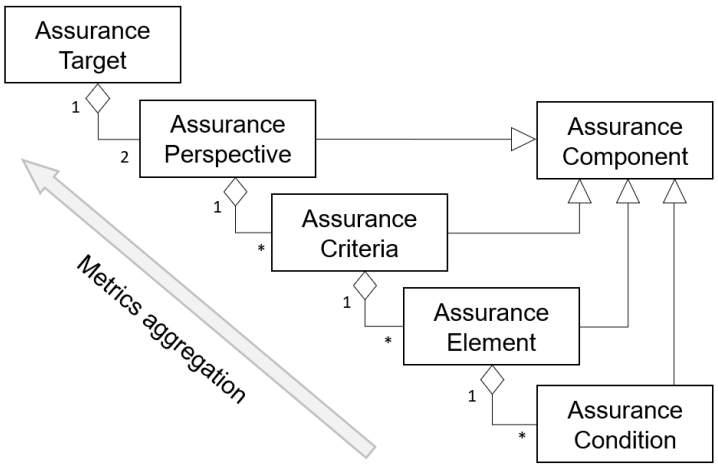

Figure 1. Compositions of the security assurance evaluation model.

The concepts for each component are described below. For simplicity of presentation, we use "assurance" in short to represent the term "security assurance" for all component names in the model.

Assurance Perspective. Assurance perspectives describe the interrelation or relative significance in which an assurance target is evaluated. In our approach, two perspectives on cyber security are taken in the evaluation: security requirements and vulnerabilities. The former addresses the positive side of system security while the latter considers the negative side that involves looking inside the system for structural flaws and weaknesses. We assume that, on the one hand, fulfilling security requirements through implementing countermeasures and checking its correct functionality will give protection against unintentional errors. On the other hand, with proper identification and addressing of vulnerabilities, it can go a long way toward reducing the probability and impact of threats materializing in the system. We argue that even if security mechanisms are properly elucidated at the requirement stage, they could result in weakness if they are inappropriately implemented or deployed. Consequently, while evaluating security assurance, security requirement improves the assurance posture; contrariwise, the existence of vulnerabilities leads to a reduction in the assurance level. Such concepts will be inherited by the rest of the assurance components.

Assurance Criteria. Assurance criteria are the specific properties that will be selected, tested, and measured to confirm the sufficiency of system security to be offered to users. The term assurance criteria as used in this model refers to a higher, more abstract level of

meaning that can be thought of as a standard in the assurance target's application domain. These criteria are part of the "target" that the work is planned to achieve (or eliminate in the perspective of vulnerabilities). In our quantitative security assurance approach, assurance criteria play an especially important role in the assurance evaluation, which provides a basis for comparison among different assurance targets; a reference point against which another system can be evaluated. In Table 1, we give exemplary criterion sets for an assurance target in the domain of web applications, in which the content is extracted from the OWASP Application Security Verification Standard (ASVS) [13] (in defining security requirement criteria) and OWASP Top 10 [24] (in defining vulnerability criteria). The former provides a rigorous list of security requirements for testers, developers, security professionals, and consumers, while the latter lists the ten most common web application security risks nowadays.

Table 1. Exemplary assurance criteria in web applications.

Security Requirement Criteria	Vulnerability Criteria
Authentication	Broken Access Control
Access Control	Cryptographic Failure
Validation, Sanitization, and Encoding	Injection
Errors, Logging, and Auditing	Security Misconfiguration
Data Protection	Identification and Authentication Failure

We argue that the foundation for quantifying and analyzing metrics in SA is to understand what "criteria" are of interest and of "how important" each is expected to be. The assurance criteria are formulated depending on the objectives and functions of the assurance target. Concerning security, not all security requirements should be treated as equally important [2]. Likewise, the vulnerabilities in need of fixing must be prioritized based on which ones pose the most immediate danger. To reflect that, one must specify a numeric factor for each assurance criterion: "Weight" for security requirement criteria and "Risk" for vulnerability criteria. On the one hand, the weighing factor emphasizes the contribution of particular aspects of security requirements over others to the security assurance result, thereby highlighting those aspects in comparison to others in the SA analysis. That is, rather than each security requirement (criteria) in the whole dataset contributing equally to the result, some of the data are adjusted to make a greater contribution than others. The weight factor expresses how security is emphasized in the assurance target and it must be caried out based on the application context. For example, if authentication is necessary to make a specific API secure, that security requirement should be given particular importance, hence the weight is also high. On the other hand, from the perspective of vulnerabilities, the term risk can be defined as the probability and the consequence of an unwanted incident caused by existing vulnerabilities. That is, a risk is an impact of uncertainty on systems, organizations, etc. Several frameworks and methods have been developed for risk analysis, and organizations may choose their method depending on the type of risks they encounter, or their business area, for example, common vulnerability scoring system (CVSS) [25] and damage, reproducibility, exploitability, affected users, and discoverability (DREAD) [26].

Assurance Element. Assurance criteria are narrated in detail by a set of assurance elements. As in assurance criteria, assurance elements are divided into security requirement elements and vulnerability elements. The former represents requirement items needed to be fulfilled, while the latter indicate a particular kind of vulnerability potentially existing in the assurance target. Table 2 lists the exemplary elements with the corresponding assurance criteria, extracted from OWASP ASVS as well.

Table 2. Exemplary assurance elements (security requirement elements).

Security Requirement Criteria	Security Requirement Element
Authentication	Password Security Credential Storage Credential Recovery One Time Verifier
Validation, Sanitization, and Encoding	Input Validation Sanitization and Sandboxing Output Encoding Deserialization Prevention

Assurance Condition. An assurance condition describes the underlying constraints (or terms) of assurance elements that need to be taken into consideration in assurance evaluation. It is specifically defined according to the organizational contexts, which include special circumstance items, such as the deployment environment, the organization's current state, and the security concerns. In addition, assurance conditions can also be represented as test cases performed to check to what extent the security requirements' conditions and the vulnerabilities' conditions are true. Table 3 represents the exemplary security requirement conditions under the element of "Password Security".

Table 3. Exemplary assurance conditions (security requirement conditions).

Security Requirement Element	Security Requirement Condition
Password Security	The passwords should be at least 64 characters, and passwords of more than 128 characters are denied. Password truncation is not performed. However, consecutive multiple spaces may be replaced by a single space. Any printable Unicode character, including language-neutral characters such as spaces and emojis, are permitted in passwords. Password change functionality requires the user's current and new password. A password strength meter is provided to help users set a stronger password.

3.2. Assurance Metrics Calculation: The Core Concept

SA evaluation is a systematic process of assigning meaningful scores to the assurance target that indicate its security posture [27]. The higher the value, the better the trustworthiness of the system product against its security mechanisms. For deriving the final score, in our approach, metrics of assurance components are computed using a bottom-up approach, which involves the estimation of at the lowest possible level of detail. It is essential to be able to define what is meant and how to measure it when SA is evaluated. We suggest an aggregation method for doing so by using the model as the structure for estimating values related to SA into a single measure. Figure 2 depicts the hierarchical structure of the SA evaluation based on the proposed SA evaluation model, while Table 4 describes each notion. Our quantitative approach divides the SA evaluation into three sequential phases: the first phase of evaluation is responsible for the assessment of the assurance elements; the second phase for the evaluation of the assurance criteria; and the third for the assurance perspectives, in turn, of the overall assurance level of the assurance target. With the term "evaluation", we refer to the assignment of a metric to each component in the model. Metrics represent measurement or evaluation indexes that are given attributes to satisfy the security assurance evaluation. The three-phase quantitative process is discussed in the following subsections.

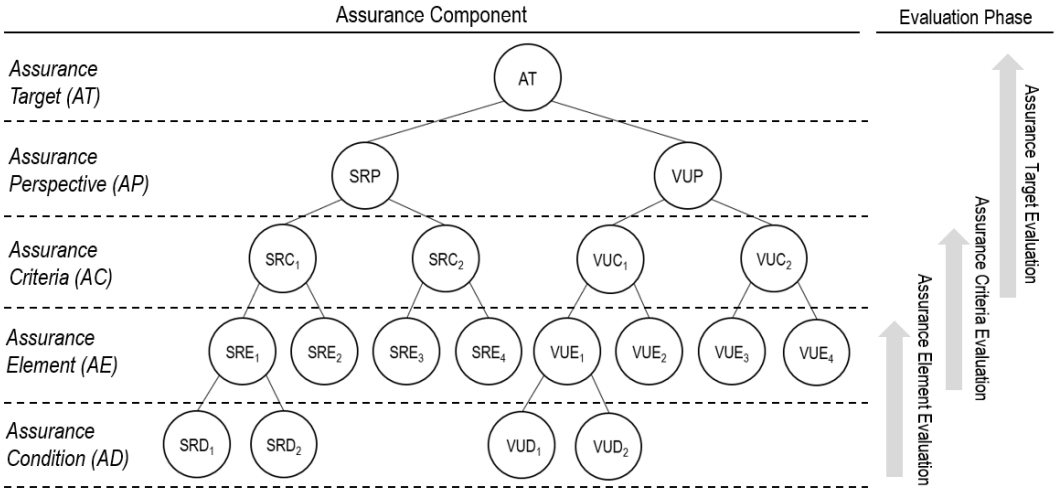

Figure 2. Sample hierarchical structure of the security assurance evaluation.

Table 4. Symbols used in the assurance evaluation.

Symbol	Description
AT	Assurance target
SRP	Security requirement perspective
VUP	Vulnerability perspective
SRC	Security requirement criteria
VUC	Vulnerability criteria
SRE	Security requirement element
VUE	Vulnerability element
SRD	Security requirement condition
VUD	Vulnerability condition

3.2.1. Assurance Element Evaluation Phase

The first phase of assurance evaluation is responsible for the assessment of the assurance elements of the SA evaluation model, from the quantification of the corresponding assurance conditions. In our test-based methodology, each assurance condition is mapped to one test case to decide fulfillment scores (for SRD) or existence scores (for VUD). For SRD, results for test cases are primarily pass or fail, where a pass indicates that the corresponding SRD is "Fully fulfilled" (fulfillment score = 1), while a failure of a test case means that the SRD is "Not fulfilled" (fulfillment score = 0). However, in some test cases, the result can be considered a 'Partial fulfillment'. Partial fulfillment means that the actual result matches its expected result, however, there are more rigorous criteria/specifications needing to be met in order to strongly claim the full fulfillment. In addition, an unnecessary (or superfluous) exception/message that is caught during the test-case execution can be also treated as a partial fulfillment [28,29]. Such a test execution state is usually applied in the context of manual testing, heavily reliant on the tester's judgment [30]. For example, it is assumed that the SHA-1 encryption algorithm is found in testing the SRD "The system stores account password in approved encrypted formats". In this case, even though there is evidence showing the password is encrypted, we see this test case to be a partial pass, as the SHA-1 is not considered a strong-enough password encryption function [31]. Therefore, the assurance score of the SRD is assigned a value of 0.5, indicating "Partially fulfilled". Similarly, the existence score for VUD has two value options, where 0 means no vulnerability indicated by the test results, and 1 represents the existence of the vulnerability.

The scores for SRE and VUE are calculated separately. For an SRE, once the fulfillment scores are decided in all associated SRDs, its score can be calculated. We define a metric *ActSRD* as a measurement to reflect the actual (calculated) score of SRE. The value of *ActSRE* is obtained by averaging the fulfillment scores of the related SRD. Since the SRDs

we add together are similar ones, by using the "Average" function, we can consider all the relevant items to derive a representative score of the whole dataset. Additionally, the assurance conditions are designed in such a way that each condition will cover one perspective of the assurance element. Failing the whole element if one condition fails is not fair for the rest of the conditions. The following formula represents the calculation of the *i*-th SRE score (represented as *ActSREi*):

$$ActSRE_i = \frac{\sum_{j=1}^{n} ActSRD_{ij}}{n}, \forall \text{ActSRD} \in \{0, 0.5, 1\}, \quad (1)$$

where:

ActSRDij: the actual (fulfillment) score of the *j*-th SRD associated with the *i*-th SRE;
n: the number of SRDs associated with the *i*-th SRE.

Similarly, the formula used for calculating the actual VUE score (*ActVUE*) is defined as the average of the corresponding VUD existence score, represented below:

$$ActVUE_i = \frac{\sum_{j=1}^{n} ActVUD_{ij}}{n}, \forall ActVUD_{ij} \in \{0, 1\}, \quad (2)$$

where:

$ActVUD_{ij}$: the existence score of the *j*-th VUD associated with the *i*-th VUC;
n: the number of VUDs associated with the *i*-th VUE.

3.2.2. Assurance Criteria Evaluation Phase

The second phase of assurance evaluation is responsible for the calculation of assurance criteria scores. Based on the previous discussion, the actual score of the *i*-th SRC, represented by *ActSRCi*, is measured based on the average value of its respective SRE and obtained by multiplying a weight factor to express the levels of importance. The scale of the weight factor ranges from 1 to 10, where 1 is assigned to SRCs that are least essential, while 10 is the maximum expressing a vital requirement. The formula to calculate *ActSRCi* is defined as:

$$ActSRC_i = WghSRC_i \times \frac{\sum_{j=1}^{n} ActSRE_{ij}}{n}, \forall WghSRC_i \in [1, 10], \quad (3)$$

where:

ActSRCij: the actual score of the *j*-th SRE associated with the *i*-th SRE;
WghSRCi: the weight factor that corresponds to the *i*-th SRC;
n: the number of SREs associated with the *i*-th SRC.

Based on Equation (3), it can be derived that *ActSRC* has a maximum value, equaling its weight factor when all the underlying security requirements are fulfilled (i.e., *ActSREi* = 1).

The assurance metric *ActVUCi*, represented by the *i*-th vulnerability criteria, can be calculated using the average value of correspondent VUEs, considering the risk factor of vulnerabilities as well. It has to be mentioned that the risk is usually derived using the standard risk model: $Risk = Likelihood \times Impact$ Risk. With this flexible model, the scale of the resulting risk value could range from 0 to 10, where 0 represents that the corresponding VUC is least likely to fail, while 10 is considered the maximum risk. The formula to derive the *i*-th *ActVUC* is defined as:

$$ActVUC_i = RskVUC_i \times \frac{\sum_{j=1}^{n} ActVUE_j}{n}, \forall RskVUC_i \in [0, 10], \quad (4)$$

where:

ActVUCij: the actual score of the *j*-th VUE associated with the i-th SRC;
RskVUCi: the risk that corresponds to the *i*-th VUC;
n: the number of VUEs associated with the *i*-th VUC.

3.2.3. Assurance Target Evaluation Phase

The third phase of evaluation is responsible for the calculation of the overall assurance score for the assurance target. This is achieved by aggregating the score of the assurance criteria and perspectives at the following three levels of calculation.

Level 1. The first level is to obtain a summative assurance score for each assurance perspective by accumulating the correspondent assurance criteria. For the assurance perspective SRP, we define a metric *ActSRP* to present the overall security requirement score of the assurance target. The formula is as follows:

$$ActSRP = \sum_{i=1}^{n} ActSRC_i, \tag{5}$$

where:

ActSRCi: the actual score of the *i*-th SRC;

n: the number of SRCs.

Correspondingly, the formula used for the calculation of the overall vulnerability score is presented below:

$$ActVUP = \sum_{i=1}^{n} ActVUC_i, \tag{6}$$

where:

ActVUCi: the assurance score of the *i*-th vulnerability criterion;

n: the number of VUCs.

Level 2. At the second level, the security assurance score (*ActSAS*) of the assurance target is derived by using the difference between the security requirement score (*ActSRP*) and vulnerability score (*ActVUP*). Thus, the formula is as follows:

$$ActSAS = ActSRP - ActVUP. \tag{7}$$

Level 3. It can be noticed that the scale of *ActSAS* is highly influenced by the number of security requirements as well as vulnerabilities included in the evaluation model (Equations (5) and (6)). This leads to a variable range of assurance scores among different assurance targets and, further, makes it difficult to interpret to make decisions among various systems. In this regard, *ActSAS* must be normalized to a common scale for a more comprehensive and understandable value, named the assurance level (*SAL*). We adopt the min–max normalization method [32], which preserves the relationships among the original data values. This method will encounter an out-of-bounds error if a future input case for normalization falls outside the first data range for the attribute. The formula of this generic normalization method is presented as follows:

$$v' = \frac{v - min_A}{max_A - min_A}(newmax_A - newmin_A) + newmin_A, \tag{8}$$

where:

min_A and max_A: the minimum and maximum values of an attribute;

$newmin_A$ and $newmax_A$ and: the new minimum and maximum values after normalization;

v: the old value of an attribute;

v': the new value after normalization.

The convention we follow for the *SAL* is that it lies in the interval between 0 and 10, where 0 corresponds to the worst possible level of security assurance, while 10 to an excellent assurance level. Thus, the formula for the metric *SAL* can be simply defined as:

$$SAL = \frac{ActSAS - MinSAS}{MaxSAS - MinSAS} \times (10 - 0) + 0 = \frac{ActSAS - MinSAS}{MaxSAS - MinSAS} \times 10. \tag{9}$$

To derive *MaxSAS*, we can refer to Equation (7), from which we know that *SAS* can be maximum when the following two conditions are met:

1. all security requirements are fulfilled, which causes the value of *ActSRP* to be maximum, and

2. all possible vulnerabilities do not exist. This makes *ActVUP* minimum (zero).

SAS, on the other hand, can become minimum if (i) all protection mechanisms are ineffective to fulfill the defined security requirements (*ActSRP* is minimum), and (ii) all listed vulnerabilities are found to exist in the assurance target, and all have maximum risk value (*ActVUP* is maximum).

3.3. More Metrics

In the previous section, we have presented the core concept of how the overall assurance score of an assurance target is calculated based on the hierarchical SA evaluation model. Calculation of these scores is the first step toward quantitative SA evaluation. In this section, we introduce more metrics to conduct a comprehensive analysis and evaluation for both perspectives of security requirements and vulnerabilities.

3.3.1. Security Requirement Metrics

Security requirement metrics relate to a measurement that evaluates whether security protection mechanisms exist and fulfill defined security requirements. We identified metrics as part of the security requirement metrics category when the metric is primarily a measure of requirements and their specification. We identified three subcategories of security requirement metrics: performance metrics, impact metrics, and prioritization metrics.

Security Requirement Performance. These metrics are used to gauge to what extent security protection mechanisms exist and fulfill defined security requirements. This performance metric is measured mainly at the level of assurance perspectives as well as assurance criteria, derived using the ratio between the actual score and maximum score. Thus, the formula to calculate the performance of the *i*-th SRC is:

$$PerSRC_i = \frac{ActSRC_i}{MaxSRC_i} \times 100\%. \tag{10}$$

Consequently, the formula to calculate the performance of the SRP is defined as follows.

$$PerSRP = \frac{\sum_{i=1}^{n} ActSRC_i}{\sum_{i=1}^{n} MaxSRC_i} \times 100\%, \tag{11}$$

where:

n: the number of SRCs.

Security Requirement Impact. These metrics are used to measure and identify the positive effects (or contribution) of security requirement fulfillment on the security assurance score. Knowing the impact will allow stakeholders to figure out ways to maximize the positive in alignment with the security goal. To measure the impact of security requirements, we adopt two categories of metrics: (1) the impact on the overall security requirement (i.e., SRP), and (2) the impact on the security assurance score (i.e., SAS). The former metrics constrain the impact evaluation within the perspective of security requirements only, while the latter expands the scope to the whole assurance score (including the vulnerability perspective). To investigate the range of the security requirement impact, for each category, we calculate its maximum and actual value. As a result, four metrics are defined to evaluate the security requirements impact. Table 5 presents the four metrics for the evaluation of the security requirement impact at the level of SRC and the corresponding formula. We also apply the impact metrics at the level of SRP, but only consider the category of "Impact to SAS". Table 6 lists the two metrics for SRP.

Table 5. Security requirement impact metrics (for SRC).

	Maximum Possible Impact of SRC		Actual Impact of SRC	
Impact to SRP	$MpiSrpSRC = \frac{MaxSRC_i}{\sum_{i=1}^{n} MaxSRC_i} \times 100\%$	(12)	$AciSrpSRC = \frac{ActSRC_i}{\sum_{i=1}^{n} MaxSRC_i} \times 100\%$	(13)
Impact to SAS	$MpiSasSRC = \frac{MaxSRC_i}{MaxSAS - MinSAS} \times 100\%$	(14)	$AciSasSRC = \frac{ActSRC_i}{MaxSAS - MinSAS} \times 100\%$	(15)

n: the number of SRCs.

Table 6. Security requirement impact metrics (for SRP).

	Maximum Possible Impact of SRP		Actual Impact of SRP	
Impact to SAS	$MpiSasSRP = \sum_{i=1}^{n} MpiSasSRC_i$	(16)	$AiSasSRP = \sum_{i=1}^{n} AciSasSRC_i$	(17)

n: the number of SRCs.

Security Requirement Prioritization. After the performance and impact of security requirements are measured through the above metrics, it is necessary for stakeholders taking action to implement corresponding security mechanisms to fulfill the expected requirements. However, with limited resources in organizations, it is difficult for security stakeholders to fulfill these security requirements simultaneously. To help stakeholders determine the order of implementation, we define a metric at the level of SRC, named Priority Score (*PrsSRC*), which is calculated using the formula as follows:

$$PrsSRC_i = \frac{MaxSRC_i - ActSRC_i}{MaxSRC_i}. \quad (18)$$

3.3.2. Vulnerability Metrics

Following the same concept practices in the development of security requirement metrics, we define a set of vulnerability metrics to evaluate the weakness of the assurance target, listed in Table 7. In contrast to the positive contribution of security requirement scores on the overall assurance score, in the perspective of vulnerabilities, the higher the actual score, the more severe an assurance component is. That means the scores of all vulnerability components always result in a negative effect on the result. In this regard, there are slight differences in formula definitions in terms of the performance, the actual impact, and the priority score of vulnerabilities.

Table 7. Summary of vulnerability metrics.

Metrics	Description	Formula	
Vulnerability Performance			
PerVUC	Performance of VUC	$\frac{MaxVUC - ActVUC_i}{MaxVUC_i} \times 100\%$	(19)
PerVUP	Performance of VUP	$\frac{\sum_{i=1}^{n} MaxVUC_i - \sum_{i=1}^{n} ActVUC_i}{\sum_{i=1}^{n} MaxVUC_i} \times 100\%$	(20)
Vulnerability Impact			
MpiVupVUC	Maximum possible impact of VUC on VUP	$\frac{MaxVUC_i}{\sum_{i=1}^{n} MaxVUC_i} \times 100\%$	(21)
AciVupVUC	Actual impact of VUC on VUP	$\frac{MaxVUC_i - ActVUC_i}{\sum_{i=1}^{n} MaxVUC_i} \times 100\%$	(22)
MpiSasVUC	Maximum possible impact of VUC on SAS	$\frac{MaxVUC_i}{MaxSAS - MinSAS} \times 100\%$	(23)
AciSasVUC	Actual impact of VUC on SAS	$\frac{MaxVUC_i - ActVUC_i}{MaxSAS - MinSAS} \times 100\%$	(24)
MpiSasVUP	Maximum possible impact of VUP on SAS	$\sum_{i=1}^{n} MpiSasVUC_i$	(25)
AciSasVUP	Actual impact of VUP on SAS	$\sum_{i=1}^{n} AciSasVUC_i$	(26)
Vulnerability Priority			
PrsVUC	Priority score of VUC	$\frac{ActVUC_i}{MaxVUC_i}$	(27)

n: the number of VUC.

4. Security Assurance Analytics

Analytics is the process of discovering, interpreting, and communicating significant patterns in data [33]. It helps people see insights that they might not otherwise detect. In this section, we present two exemplary instruments for security assurance analytics: a security assurance scorecard and a dashboard. It is important to note that these examples are for an illustrative and demonstrative purpose, and do not rely on specific dashboard tools—nor do they require the use of (programming) technologies. The goal of these examples is to present how the proposed assurance metrics dataset may be used in security assurance analytics to support the decision-making process.

4.1. Security Assurance Scoreboard

A scorecard is a report technique that allows management to measure the achievement of current activities of the organization in comparison to planned goals or outcomes [34], which includes a set of statistical records and meanwhile employs a top-down approach to present the indicators. Figures 3–5 demonstrate exemplary assurance scoreboards filled with our proposed assurance metrics. In this study, an evaluation result of an open-source web-based issue management system is presented. This system allows project members to document and track project issues as well as to generate management reports. For the SA evaluation to work, we leveraged OWASP as the knowledge source to prepare the content of assurance components in the domain of web applications. Three OWASP project materials were chosen: OWASP ASVS, OWASP Top 10, and OWASP Web Security Testing Guide (WSTG) [35]. The first material was used to construct a set of security requirements, while the last two were synthesized for the vulnerability assessment. As a result, a total of eight security requirements and eight vulnerabilities are considered crucial for evaluating the assurance target.

	Max.	Actual
Security assurance score	61.00	43.62
Security assurance level	10	7.9

	Score		Impact on SAS		
Assurance perspective	**Max.**	**Actual**	**Max.**	**Actual**	**Performance**
Security requirement	61.00	46.48	73.76%	56.20%	76.20%
Vulnerability	21.70	2.86	26.24%	22.79%	86.84%

Figure 3. Security assurance scoreboard—top level.

								SRC Score		Impact on SRP		Impact on SAS			Priority	
Security Requirement Criteria	**Wgh**	**SRE**	**FF**	**PF**	**NF**	**Sum**	**Avg**	**Max**	**Actual**	**Max**	**Actual**	**Max**	**Actual**	**Perf**	**Score**	**Rank**
Authentication	9	10	5	3	2	8.20	0.82	9.00	7.38	14.75%	12.10%	10.88%	8.92%	82.00%	0.180	7
Session Management	8	7	5	1	1	5.40	0.77	8.00	6.17	13.11%	10.12%	9.67%	7.46%	77.14%	0.229	5
Access Control	9	8	5	1	1	6.80	0.85	9.00	7.65	14.75%	12.54%	10.88%	9.25%	85.00%	0.150	8
Stored Cryptography	6	5	2	1	2	2.88	0.58	6.00	3.46	9.84%	5.67%	7.25%	4.18%	57.60%	0.424	1
Error Handling and Logging	7	5	3	1	1	3.88	0.78	7.00	5.43	11.48%	8.90%	8.46%	6.57%	77.60%	0.224	4
Data Protection	9	3	1	2	0	2.28	0.76	9.00	6.84	14.75%	11.21%	10.88%	8.27%	76.00%	0.240	3
API and Web Service	6	6	2	3	1	3.95	0.66	6.00	3.95	9.84%	6.48%	7.25%	4.78%	65.83%	0.342	2
Configuration	7	5	4	0	1	4.00	0.80	7.00	5.60	11.48%	9.18%	8.46%	6.77%	80.00%	0.200	6

Wgh Weight
SRE Number of SREs
Perf Performance
FF Number of full-fulfillment SREs
PF Number partial-fulfillment SREs
NF Number of nonfulfillment SREs

Figure 4. Security assurance scoreboard—security requirement metrics.

Vulnerability Criteria	Lk	Imp	Rsk	VUE	EE	NE	Sum	Avg	VUC Score		Impact on VUP		Impact on SAS		Perf	Priority Score	Rank
									Max	Actual	Max	Actual	Max	Actual			
Broken Access Control	0.038	5.93	2.25	4	1	3	0.75	0.19	2.25	0.42	10.38%	1.95%	2.72%	2.21%	81.25%	0.188	3
Cryptographic Failures	0.045	6.81	3.06	5	3	2	1.35	0.27	3.06	0.83	14.12%	3.81%	3.71%	2.71%	73.00%	0.270	1
Injection	0.034	7.15	2.43	13	3	10	1.25	0.10	2.43	0.23	11.20%	1.08%	2.94%	2.66%	90.38%	0.096	5
Security Misconfiguration	0.045	6.56	2.95	8	2	6	1.67	0.21	2.95	0.62	13.60%	2.84%	3.57%	2.82%	79.13%	0.209	2
Vulnerable and Outdated Components	0.088	5	4.40	3	0	3	0.00	0.00	4.40	0.00	20.27%	0.00%	5.32%	5.32%	100.00%	0.000	8
Identification and Authentication Failures	0.026	6.5	1.69	19	2	17	1.00	0.05	1.69	0.09	7.79%	0.41%	2.04%	1.94%	94.74%	0.053	7
Software and Data Integrity Failures	0.021	7.94	1.67	9	1	8	0.67	0.07	1.67	0.12	7.68%	0.57%	2.02%	1.87%	92.56%	0.074	5
Security Logging and Monitoring Failures	0.065	4.99	3.24	4	1	3	0.67	0.17	3.24	0.54	14.95%	2.50%	3.92%	3.26%	83.25%	0.168	4

Lk Likelihood **VUE** Number of VUEs
Imp Impact **EE** Number of existent VUEs
Rsk Risk **NE** Number of non-existent VUEs
Perf Performance

Figure 5. Security assurance scoreboard—vulnerability metrics.

In Figure 3, the "top-level" scoreboard structure is presented, which comprises assurance metrics showing the summarized SA evaluation result of the assurance target, including the overall assurance score, the assurance level, and the assurance metrics in the perspective of the security requirement and the vulnerability. These data provide a "panoptic view" by aggregating the metrics from both sets of assurance perspectives (security requirements and vulnerabilities). For a better comparison of metrics, the corresponding maximum and actual values are listed side-by-side. From Figure 3, we can see that the security requirement perspective has a major role in the overall SA evaluation since it achieves up to 73.76% of the assurance score, while the vulnerability perspective, on the other hand, contributes only 26.24%. Furthermore, in terms of performance, the security requirements generally gain a worse score than vulnerabilities (76.20% vs. 86.84%). Thus, it is suggested that security requirements should be treated as higher priorities while taking improvement actions.

Figures 4 and 5 are two drill-down scoreboards providing the "next-level" details of security requirement metrics and vulnerability metrics. The two tables permit metrics to be monitored through the level of assurance criteria, which allows an effective and "tidy" collection of the stakeholders' views of the areas of interest. In that table, rows identify the criteria, while the columns identify the areas of interest and the key assurance metrics. It is noted that, in Figure 4, the weight factors for all security requirement criteria are estimated by security and domain experts using a subjective weighting approach. As we can see, among the eight security requirement criteria, "Access Control" has the highest actual impact on security assurance score calculation (i.e., 9.25%) as well as the best performance (85%). On the other hand, "Stored Cryptography" is the worst assurance criterion since both the performance and the impact on SAS are the lowest (57.6% and 4.18%, respectively).

According to the vulnerability metrics shown in Figure 5, the criterion "Vulnerable and Outdated Components" has a maximum impact on security assurance score, meanwhile, it reaches an outstanding performance (100%). As the poor performance of "Cryptography" is identified in the security requirement perspective, the corresponding weakness in terms of the "Cryptographic Failures" is also disclosed in the vulnerability perspective (performance: 73%).

While assigning the risk factor for each vulnerability criterion, we consider the standard risk model:

$$Risk = Likelihood \times Impact.$$

In this case, we adopt the data factors defined in OWASP Top 10 categories [36], which are systematically derived using CVSS v3. To calculate the corresponding risk of an OWASP Top 10 vulnerability, we take the data factor "Average Incidence Rate" as the *likelihood*, while *impact* uses "Average Weighed Impact". Table 8 shows the snapshot data factors of the vulnerability "Broken Access Control" in OWASP Top 10. According to the table, the initial risk factor of the vulnerability is calculated as 3.18% $\times$ 5.93 = 0.225. After the risk

value is derived, we also place it on a 0–10 scale, with the same range as the weighting factor. As a result, the final risk factor becomes 2.25.

Table 8. Data factors of "Broken Access Control" in OWASP Top 10.

Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Max Coverage	Avg Coverage	Total Occurrences
55.97%	3.81%	6.92	5.93	94.55%	47.72%	318,487

Except for the metrics introduced in the previous sections, several new ones are added to the scorecard to enhance the analysis capabilities, including the security requirement fulfillment metrics (divided into full, partial, and weak fulfillment), vulnerability existence metrics (divided into existence and non-existence), and the ranking numbers. The ranking numbers are generated following the ascending order of the criteria. As we can see in the rank, the security requirement "Stored Cryptography" has the highest priority, meaning it does not perform very well and is currently the most crucial in security assurance evaluation, concluded in line with our previous analysis. Therefore, the stakeholders should emphasize more on this criterion to improve the security level of the assurance target.

4.2. Security Assurance Analytics Dashboard

To advance SA analysis, another technique for presenting the assurance metrics is the "dashboard". Dashboards are a collection of graphs, charts, gauges, or other visual representations that serve the purpose of viewing multiple datasets at a time [37]. Such visualizations provided by dashboards can lead to the more clear identification of previously unnoticed patterns in data, informing improvement initiatives, and more efficient and effective decision making [38]. Analytics dashboards are typically used to compile large volumes of complex data into measurable key performance indexes (KPIs) that allow businesses to understand and infer meaningful insights. Figure 6 demonstrates an illustrated SA analytics dashboard for the single system analysis, filled with the proposed metrics dataset. This SA analytics dashboard presents a bird's eye view of an assurance target's overall performance, which simplifies the metrics data into more manageable chunks of visual information that allows security stakeholders to easily oversee and explore what they are doing right and what needs to improve. For an effective KPI presentation, the proposed SA dashboard is split into three rows, each containing more reporting panels, starting with a high-level overview and providing easy paths for users to increase the level of granularity.

In the first row (Figure 6a), we select and organize the most critical metrics based on what the audience should be notified of at first sight, supporting them by answering the following questions:

- What is the security assurance level of the assurance target?
- What is the overall security performance in the aspects of the security requirement implementation and the vulnerability mitigation?
- To what extent are security requirements fulfilled and the number of vulnerabilities found?

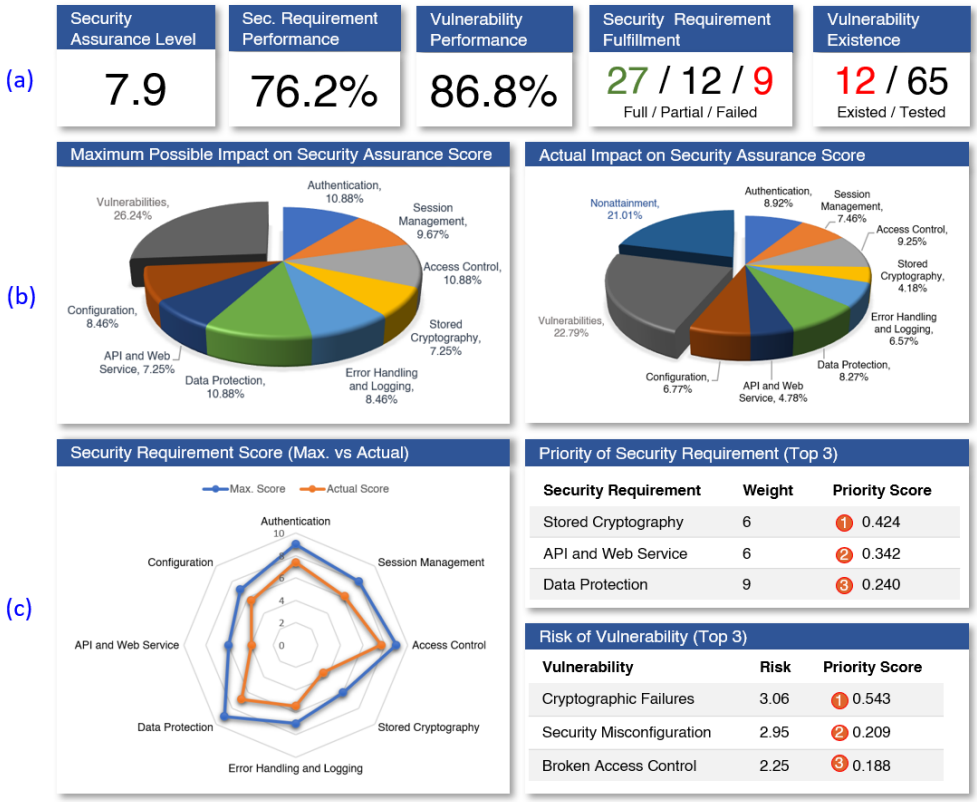

Figure 6. Security assurance analytics dashboard—single system view.

In the middle row of the dashboard (Figure 6b), the impact of security requirements and vulnerabilities on the security assurance score is presented, distinguished by two reporting panels: the maximin and actual impact. To represent the metrics in a pie chart, we expand the security requirement perspective to the next level (i.e., security requirement criteria), while the vulnerabilities are left as a summary item. This design allows better recognition of the distribution of the impact of security requirements and, at the same time, provides comparative information between the maximum and actual value. In addition, in the actual impact pie chart (the right-side panel), the total occupancy of the non-attainment in security requirements and vulnerabilities is highlighted (i.e., 21.01%). This is the area leading to the credit loss in the security assurance evaluation, which illustrates the weakness portion of the assurance target.

The lower row (Figure 6c) is divided into two separated panels. The left panel provides a comparative analysis of security requirement criteria. The use of a radar chart is useful for seeing which criteria are scoring high or low, and the variation between the maximum and actual value. The right-side panels identify the top three critical security requirements and vulnerabilities that the security stakeholders should assign high priorities to improve the security assurance score.

Apart from analyzing the metrics for a single system, another aspect to consider in SA analytics is multiple-system comparison. Figure 7 presents a format of the SA analytic dashboard which provides the capability of assurance metrics comparison between multiple systems. Such a dashboard serves as a comparative platform that consumers can utilize to investigate the difference in the relative score (or performance) between similar ICT systems when they have slightly different features but are within the same domain, e.g., web applications. Since this dashboard attempts to collate and compare quantitative metrics from different systems, it is necessary, to make any of the views comparable, to have a methodology that narrows the collection of views to very specific components. This is achieved by selecting the most critical metrics and using a consistent color-coding scheme to give a visual aid in the identification of different systems and assurance metrics.

Figure 7. Security assurance analytics dashboard—multiple systems view.

5. Conclusions

This paper aims to develop a novel approach for the modeling, calculation, and analysis of SA metrics that could ultimately enhance quantitative SA evaluation. This paper first presents a modeling approach to structuring the SA component for the quantitative SA evaluation, which is followed by a description of a comprehensive set of SA metrics with corresponding calculation algorithms. We also show the way to use the metrics by demonstrating examples of SA analytics, to gauge the confidence of the deployed security mechanisms on the assurance target.

At one level, in terms of a methodological approach to modeling the SA components and calculating metrics, this paper has provided some indication that it has value. First, to achieve the reusability of the SA evaluation models as well as flexibility in SA score calculation, the model is designed in a sufficiently generic fashion that can be applied to any application domain, regardless of the subject of the evaluation. Second, the approach considers the adaptivity and accuracy of SA metrics regarding the application domain and the organizational context. To address this, our model considers the context of the intended environment (i.e., assurance conditions) that the assurance target operates in. We conceive the SA metrics as the aggregated value of the score of contextual conditions that are directly quantified from the test results. In addition, assessing scores for test results of assurance conditions and aggregating these scores to the corresponding assurance components is easier than directly finding a single score for them.

In terms of SA analysis, except for measuring the basic score of the assurance components, we make further steps in the evolution of metrics toward better SA evaluation and analysis. For example, the impact of security requirements (or vulnerabilities) on the overall assurance score, and the recommendation on the prioritizing improvement. Knowing the impact will allow stakeholders to figure out ways to maximize the positive in alignment with the security goal, while prioritization allows the identification of critical areas that require work, addressing organizational concerns, and improving the ability to properly allocate resources. The presentations and analyses of the metrics are then demonstrated using two illustrated analytics: the SA scoreboard and the SA dashboard. The proposed SA analytics focus on using the meaningful information derived from the assurance metrics to know more about the security posture of the assurance target and, further, make informed decisions to improve the security performance. As shown by the examples, our proposed assurance metrics provide a picture of clear measurements of the key components of security assurance as well as an overall score of an assurance target. In addition, the strength and weaknesses of security requirements and vulnerabilities are appropriately quantified that can be used to support improved decision making and strategic planning initiatives.

With that being said, there are some limitations to the work presented here that will have to be addressed in future work. Firstly, the presented SA evaluation model in this article does not consider the interdependency among different assurance criteria, for example, authentication vs. session management in the perspective of security requirement, and broken access control vs. injection in the vulnerability perspective. In addition, it does not discuss the possible interaction between security requirements and vulnerabilities that might result in a double count of the assurance score calculation. To mitigate such limitations, we put forward a restricted test-case design, in which the test cases for security requirements and vulnerabilities are made into separate datasets. Under this *prerequisite*, the two sets have no element in common but complement each other to form a completed dataset. In more depth, the security requirement testing is designed to verify compliance with security controls, asserting the expected security functionality. For example, when the application deals with personally identifiable information (PII) and sensitive data, the test case to be validated is the compliance with the company information security policy requiring encryption of such data in transit and storage. On the other hand, vulnerability testing (or penetration testing) is designed to identify gaps in security control, driven by risk, which validates the application for unexpected behavior. For example, the test case to be validated for password security could be "Verifying the authentication can be broken through a brute force or dictionary attack of passwords and account harvesting vulnerabilities in the application". Nevertheless, to improve the precision of the assurance score calculation, our future work will concentrate on modeling constraints and interdependent criteria (e.g., how the score of a criterion affects the effectiveness/correctness of others) as well as the score deduction algorithm between security requirements and associated vulnerabilities. Secondly, to assess the usability of the SA metrics and analytics reported here, there is a need to verify them on a more representative sample of security stakeholders in the SA domain. Lastly, future development of this work should also include a comparison of our proposal with more systematic approaches to categorical data collection and analysis, for instance, with business intelligence (BI). This is an important step to make as it will also allow us to improve the methodological efficacy of this approach.

Overall, it can be said that this quantitative SA approach offers the potential for disclosing more informative content to the management concerning SA evaluation. This can only be seen as yet another positive shift in the iterative process of developing and shaping the security posture and provision of metrics and analytics for appraising the system security.

Author Contributions: Conceptualization, S.-F.W., A.S., and B.K.; methodology, S.-F.W., A.S., and B.K.; development: S.-F.W.; analysis and evaluation: S.-F.W.; writing—original draft preparation, S.-F.W., writing—review and editing, S.-F.W., A.S., and B.K.; visualization, S.-F.W.; supervision, B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ouedraogo, M.; Savola, R.M.; Mouratidis, H.; Preston, D.; Khadraoui, D.; Dubois, E. Taxonomy of quality metrics for assessing assurance of security correctness. *Softw. Qual. J.* **2013**, *21*, 67–97. [CrossRef]
2. Katt, B.; Prasher, N. Quantitative security assurance. In *Exploring Security in Software Architecture and Design*; IGI Global: Hershey, PA, USA, 2019; pp. 15–46.
3. Ross, R.S. *Managing Information Security Risk: Organization, Mission, and Information System View*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2011.

4. Shukla, A.; Katt, B.; Nweke, L.O.; Yeng, P.K.; Weldehawaryat, G.K. System Security Assurance: A Systematic Literature Review. *arXiv* **2021**, arXiv:2110.01904. [CrossRef]
5. Gritzalis, D.; Karyda, M.; Gymnopoulos, L. Elaborating quantitative approaches for IT security evaluation. *Secur. Inf. Soc.* **2002**, *86*, 67–77.
6. Weldehawaryat, G.K.; Katt, B. Towards a quantitative approach for security assurance metrics. In Proceedings of the 12th International Conference on Emerging Security Information, Sochi, Russia, 12–15 September 2019.
7. Katt, B.; Prasher, N. Quantitative security assurance metrics: REST API case studies. In Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings, Madrid, Spain, 24–28 September 2018; pp. 1–7.
8. Herrmann, D.S. *Using the Common Criteria for IT Security Evaluation*; Auerbach Publications: Boca Raton, FL, USA, 2002.
9. Zhou, C.; Ramacciotti, S. Common criteria: Its limitations and advice on improvement. *Inf. Syst. Secur. Assoc. ISSA J.* **2011**, *9*, 24–28.
10. Ekclhart, A.; Fenz, S.; Goluch, G.; Weippl, E. Ontological mapping of common criteria's security assurance requirements. Proceedings of IFIP International Information Security Conference, Boston, MA, USA, 14 May 2007; pp. 85–95.
11. McGraw, G.; Chess, B.; Migues, S. *Building Security in Maturity Model*; Fortify & Cigital: Mountain View, CA, USA, 2009.
12. OWASP. Software Assurance Maturity Model v2.0. Available online: https://www.opensamm.org/ (accessed on 30 April 2022).
13. OWASP. OWASP Application Security Verification Standard (Version 4.0.3). Available online: https://owasp.org/www-project-application-security-verification-standard/ (accessed on 26 January 2022).
14. OWASP. OWASP Foundation. Available online: https://www.owasp.org/ (accessed on 30 January 2022).
15. Liu, Y.-L.; Jin, Z.-G. SAEW: A security assessment and enhancement system of Wireless Local Area Networks (WLANs). *Wirel. Pers. Commun.* **2015**, *82*, 1–19. [CrossRef]
16. Agrawal, A.; Alenezi, M.; Khan, S.A.; Kumar, R.; Khan, R.A. Multi-level fuzzy system for usable-security assessment. *J. King Saud Univ. -Comput. Inf. Sci.* **2019**, *34*, 657–665. [CrossRef]
17. Pham, N.; Riguidel, M. Security assurance aggregation for it infrastructures. In Proceedings of the 2007 Second International Conference on Systems and Networks Communications (ICSNC 2007), Cap Esterel, France, 25–31 August 2007; p. 72.
18. Ouedraogo, M.; Mouratidis, H.; Khadraoui, D.; Dubois, E. Security assurance metrics and aggregation techniques for it systems. In Proceedings of the 2009 Fourth International Conference on Internet Monitoring and Protection, Venice, Italy, 24–28 May 2009; pp. 98–102.
19. Ouedraogo, M. Towards security assurance metrics for service systems security. Proceedings of International Conference on Exploring Services Science, Berlin, Germany, 15 February 2012; pp. 361–370.
20. Rodes, B.D.; Knight, J.C.; Wasson, K.S. A security metric based on security arguments. In Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics, Hyderabad, India, 3 June 2014; pp. 66–72.
21. Heyman, T.; Scandariato, R.; Huygens, C.; Joosen, W. Using security patterns to combine security metrics. In Proceedings of the 2008 Third International Conference on Availability, Reliability and Security, Barcelona, Spain, 4–7 March 2022; pp. 1156–1163.
22. Fernandez, E.B.; Yoshioka, N.; Washizaki, H.; VanHilst, M. Measuring the level of security introduced by security patterns. In Proceedings of the 2010 International Conference on Availability, Reliability and Security, Krakow, Poland, 15–18 February 2010; pp. 565–568.
23. Villagrán-Velasco, O.; Fernández, E.B.; Ortega-Arjona, J. Refining the evaluation of the degree of security of a system built using security patterns. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Virtual, 25–28 August 2020; pp. 1–7.
24. OWASP. OWASP Top 10—2021. Available online: https://owasp.org/Top10/ (accessed on 26 January 2022).
25. Forum of Incident Response and Security Teams (FIRST). CVSS. Available online: https://www.first.org/cvss/ (accessed on 30 January 2022).
26. Burns, S.F. Threat modeling: A process to ensure application security. In *GIAC Security Essentials Certification (GSEC) Practical Assignment*; SANS Institute: Philadelphia, PA, USA, 2005.
27. Ouedraogo, M.; Khadraoui, D.; Mouratidis, H.; Dubois, E. Appraisal and reporting of security assurance at operational systems level. *J. Syst. Softw.* **2012**, *85*, 193–208. [CrossRef]
28. Arindaeng, K.; Laboriante, A.; Lu, Z.J.; Ragavendran, V. Indoor UAV Tracking System. Available online: https://azkevin.github.io/U-TRACKR/pdf/U-TRACKR.pdf (accessed on 23 April 2022).
29. Bosch, J.; Chiang, H.-F.; Gower, M. LDM-503-2 (HSC Reprocessing) Test Repor. Available online: https://dmtr-51.lsst.io/DMTR-51.pdf (accessed on 31 July 2022).
30. Reddy, N. An Excellent Compilation of Software Testing Concepts (Manual Testing). Available online: http://www.softwaretestinggenius.com/download/mtnarsir.pdf (accessed on 23 April 2022).
31. Mirante, D.; Cappos, J. *Understanding Password Database Compromises*; Technical Report TR-CSE-2013-02; Deptarthment of Computer Science and Engineering Polytechnic Institute of NYU: New York, NY, USA, 2013.
32. Jayalakshmi, T.; Santhakumaran, A. Statistical normalization and back propagation for classification. *Int. J. Comput. Theory Eng.* **2011**, *3*, 1793–8201.
33. Davenport, T.H. Analytics 3.0. *Harv. Bus. Rev.* **2013**, *91*, 64–72.
34. Kaplan, R.S.; Norton, D.P. The balanced scorecard: Measures that drive performance. *Harv. Bus. Rev.* **2005**, *83*, 172.

35. OWASP. OWASP Web Security Testing Guide. Available online: https://owasp.org/www-project-web-security-testing-guide/ (accessed on 26 January 2022).
36. OWASP. OWASP Top10 Introduction. Available online: https://owasp.org/Top10/A00_2021_Introduction/ (accessed on 27 April 2022).
37. Janes, A.; Sillitti, A.; Succi, G. Effective dashboard design. *Cut. IT J.* **2013**, *26*, 17–24.
38. Vessey, I. Cognitive fit: A theory-based analysis of the graphs versus tables literature. *Decis. Sci.* **1991**, *22*, 219–240. [CrossRef]