Towards Security Effectiveness Measurement utilizing Risk-Based Security Assurance

Reijo M. Savola, Heimo Pentikäinen
VTT Technical Research Centre of Finland
Oulu, Finland
{Reijo.Savola, Heimo.Pentikainen}@vtt.fi

Moussa Ouedraogo
Public Research Center Henri Tudor
Kirchberg, Luxembourg
Moussa.Ouedraogo@tudor.lu

***Abstract*—Systematic and practical approaches to risk-driven operational security evidence help ensure the effectiveness and efficiency of security controls in business-critical applications and services. This paper introduces an enhanced methodology to develop security effectiveness metrics that can be used in connection with correctness assurance of security controls. This methodology is then applied to an example system: a Push E-mail service. The methodology is based on threat and vulnerability analysis, and parallel security requirement and system architecture decomposition.**

Keywords-security metrics; security effectiveness metrics; security assurance level; security measurement

I. Introduction

Service-centric systems are becoming exposed to an increasing number of security threats. They also have a large number of vulnerabilities as the systems become more and more complex and connected, and market demands do not allow sufficient time for testing. Systematically obtained evidence of these systems' security performance and level is clearly beneficial for their maintenance and operation.

This paper's main contribution is in its application of a security metrics development approach, developed in earlier work by the same authors to the context of security assurance. This approach enables the measurement of security control effectiveness as a higher-level activity around the assurance of security control correctness. The study introduces a fusion of the earlier work on security metrics development [1] carried out in GEMOM (Genetic Message Oriented Secure Middleware) EU FP7 project [2] and the security assurance level measurement approach of the BUGYO Beyond (Building Security Assurance in Open Infrastructures – Beyond) CELTIC Eureka project [3]. The latter methodology builds upon well-known security assurance standards, such as the Common Criteria (CC) [4]. An example system – the Push E-mail service – investigates the utilization of the proposed methodology.

Section II discusses the background and some key concepts, Section III proposes the new methodology, and Section IV introduces the example system and discusses each step of the methodology in the context of the system. Section V presents related work, before Section VI offers some conclusions and poses some future research questions.

II. Background

A. Security Metrics in General

The term (information) *security metrics* has become standard when referring to the security level, security performance, security indicators or security strength of a System under Investigation (SuI) [5] – a technical system, product, service or organization. In this context, the term *metrics* is misleading because it implies that traditional concepts in metrology, as used in physics and other areas of science and technology, apply equally to Information Technology [6]. The complexity, lack of common definitions and dynamic nature of security risks make it impossible to measure security as a universal property. Consequently, terms such as *indicators* or *strength* might be more appropriate in the case of security-related objectives. This study, however, uses the most widely-used term, *metrics*.

Examples of security metrics application areas include risk management, comparison of security solutions, (software) security assurance, security testing, and security monitoring [7].

The ultimate goal of security measurement is to be able obtain evidence about the operational security level and performance from the SuI. Indirect security mechanisms, such as secure development processes and end-user security awareness and behavior have a remarkable impact on operational security. This impact is a deciding factor when assessing the effect that indirect security mechanisms have on operational security. Having said that, the above-mentioned mechanisms are not within the scope of this study.

There are three fundamental objectives of security measurement: (*i*) correctness, (*ii*) effectiveness, and (*iii*) efficiency of deployed security controls. Of these, effectiveness is obviously the main goal of security activities. A balanced tradeoff is needed between effectiveness and efficiency. Correctness of security controls is a necessary *but not sufficient requirement* for effectiveness. Efficiency measurement is not within the scope of this study. Information Technology Security Evaluation Criteria (ITSEC) [8] originally made the distinction between correctness and effectiveness assurance.

Table I summarizes some of the key concepts discussed in this study, with references.

978-1-4244-5494-5/10/$26.00 ©2010 IEEE

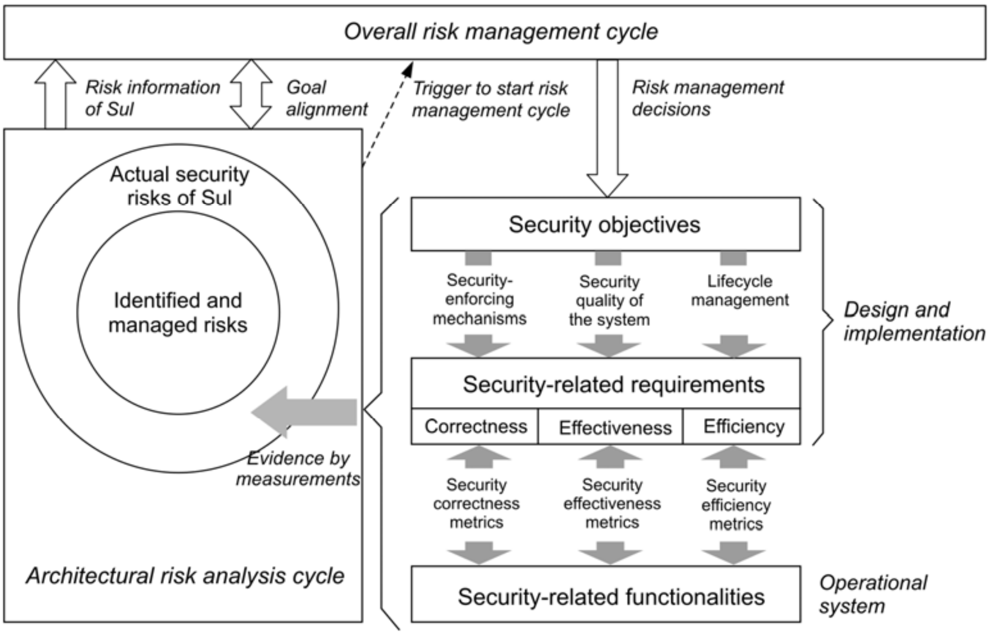
Overall risk management cycle
Risk information of SuI
Goal alignment
Trigger to start risk management cycle
Risk management decisions
Actual security risks of SuI
Identified and managed risks
Evidence by measurements
Architectural risk analysis cycle
Security objectives
Security-enforcing mechanisms
Security quality of the system
Lifecycle management
Design and implementation
Security-related requirements
Correctness
Effectiveness
Efficiency
Security correctness metrics
Security effectiveness metrics
Security efficiency metrics
Security-related functionalities
Operational system

Figure 1. The role of risk management in security objectives, requirements, functionalities and security measurement.

TABLE I. SOME KEY CONCEPTS

Concept	**Explanation**	**Ref.**
Security objective	High-level statements of intent to counter identified threats and/or satisfy identified organizational security policies and/or assumptions.	[4]
Security requirement	Requirement, stated in a standardized language, that is meant to contribute to achieving the security objectives.	[4]
Security-related requirement	Requirement, which is either a security requirement or a system design requirement, that has an impact on the security level or performance.	-
Security control	Means of managing risk, which can be administrative, technical, management, or legal in nature.	[9]
Security-related functionality	Operational functionality of the SuI that has an effect on security, including the functionality of security controls.	-
Security assurance	Grounds for confidence that an entity meets its security objectives.	[10], [11]
Confidence	Belief that an entity meets its security objectives.	[12]
Trust value of an object	Value of the belief or subjective probability of the trustor that a trustee will adequately carry out security assurance of an object.	-
Security correctness	Assurance that security controls have been correctly implemented in the SuI, and the system, its components, interfaces, and the processed data meet the security requirements.	[5], [6], [8]
Security effectiveness	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 in any way other than what is intended.	[5], [6], [8]
Security efficiency	Assurance that the adequate security quality has been achieved in the SuI, meeting the resource, time and cost constraints.	[5]

B. Operational Security Assurance and Risk Analysis

Modeling and verifying the security behavior of a realistic software-intensive or telecommunication system requires an excessive amount of time and effort. Moreover, this kind of rigorous approach is unable to track the dynamics of security risks. This creates the need for methods that can offer sufficient and credible confidence of that behavior are needed. Security assurance is defined here, according to CC, as the *grounds for confidence that an entity meets its security objectives* [10][11]. It is necessary to obtain adequate assurance that the security controls are deployed in a correct way during operation [13].

Security cannot be well managed without proper Risk Management (RM). A sub-activity risk analysis refers to the identification and ranking of risks. The effectiveness of security controls is determined by continuous risk analysis, reacting to the results in RM and updating the security objectives accordingly. At the architectural level, operational risk analysis can be implemented by Architectural Risk Analysis [14], which is sometimes referred to as threat modeling [15] or security design analysis. Based on the resulting ranked risk information, business stakeholders can make informed decisions to cancel, mitigate or accept the security risks of the SuI as a part of the overall RM process. The iteration cycle of architectural risk analysis is faster than the organization-level overall RM process and offers input to the latter. Therefore, architectural risk analysis – a necessity [14] – can trigger the overall RM cycle. In this case, decisions from the overall RM are expected as a minimum. The goals of both activities should be kept aligned. Figure 1 illustrates the role of overall RM and architectural risk analysis and shows various security metrics categories. Security objectives, the reference of security assurance practices, are identified based on the RM's decisions and other operational and compliance objectives.

978-1-4244-5494-5/10/$26.00 ©2010 IEEE

C. Relationship between Security and Security Assurance

The presence of security controls within the SuI does not mean it is always secure. In general, security assurance involves probing the specified security controls to provide information on their correct deployment and posture and their ability to mitigate or cancel risks to the system, provided that there is an adequate RM in place. The lack of assurance should raise security concerns, whether it is an absence of investigation of the security controls or because the gathered evidence creates skepticism in terms of the safeguarding measures' ability provide adequate protection. On the other hand, the lack of security controls or the knowledge that they are somehow faulty brings no assurance. However, high assurance does not automatically imply high security. Security correctness assurance observes the deployed security measures, but can prove only that the security controls have been implemented to their predefined tasks. Effectiveness measurement can only be appraised with sufficient knowledge about the threats and vulnerabilities. Given the highly dynamic nature of these elements, effectiveness can only be relative and depends heavily on the quality of RM. As the security assurance level increases, the security level of the system will also increase asymptotically, up to a limit that reflects the security quality level set by the security objectives (which, in turn, are controlled by RM). Figure 2 illustrates this relationship.

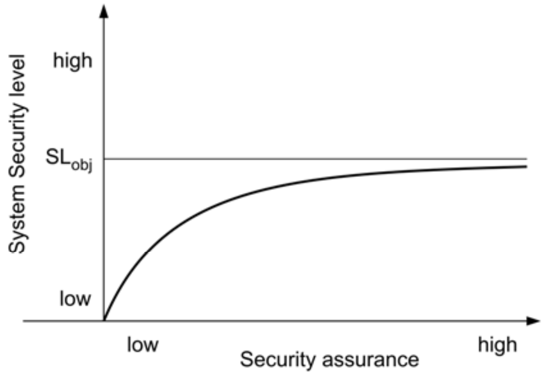

Figure 2. Linking security level of SuI and security assurance level. SL_{obj} denotes the security level set by the security objectives.

D. Security Requirements and Security-related Requirements

According to McGraw [14], design flaws account for half of all security problems. Therefore, it is important to pay attention to proper design of the SuI, manifested by a concise collection of system design requirements. In order to support RM effectively, security requirements from this collection should be identified and tagged. In addition, there are often requirements that are not directly security requirements but are relevant to security. Security-related requirements in this context refer to the subset of SuI requirements that have an impact on the security level or performance. Many other requirements (such as software quality requirements) enable security, and some (like privacy) depend on security requirements. Although this study does not address software quality in general, it must be noted that it is of utmost importance to the security level of the SuI. Security requirements are initially specified during the Research and Development (R&D) of the SuI, based on the security objectives including RM decisions. During the operational security assurance, the changes in security objectives should be reflected to the requirements. Security assurance levels address the assurance that security implementation in the operational SuI corresponds to the security requirements. Security requirements can address security functions (such as confidentiality, integrity, availability, authentication, authorization and non-repudiation), the security quality of the system, and/or system management during its lifecycle. Effective security requirements cover both overt functional security and risk-driven emergent characteristics, captured via abuse cases or attack patterns [14], resulting from operational architectural risk analysis.

E. Managed/Unmanaged Objects and Trust Values

From an operational security assurance perspective, objects of the SuI system architecture are either *managed* or *unmanaged* [10]. Security metrics are continuously used to monitor a managed object for the definition of Assurance Level for that object; this is not possible for an unmanaged one. In the latter case, there is a certain amount of trust that the security level of the object is at an adequate level. This trust can be based on, for example, the assurance claims carried out by a representative of the unmanaged object or a third party. It can be also based on reputation parameters. To illustrate this, *trust value* can be associated to a component, a ranking on the same scale as the used confidence values. This can be, for example, a number between 0.0 and 1.0, where 0.0 represents complete distrust and 1.0 represents complete trust. Trust values can also be used in managed objects; when, for example, there is no need for explicit assurance information. As with the system architecture components, it is possible to attach a trust value to security requirements, or part thereof.

III. Proposed Methodology

A. Security Metrics Development Methodology

The following iterative risk-driven methodology is proposed here for security metrics development for the context of security assurance, based on the earlier work in [1]:

1. Carry out security threat and vulnerability analysis of SuI. This step is highly iterative and can be completed as part of the *architectural risk analysis*;
2. Define measurement architecture and mechanisms for evidence collection. Carry out this stage in parallel with all the other stages in an iterative manner;
3. Utilize suitable security and system taxonomies and/or ontologies to further plan the measurement objectives and metrics types;
4. Based on the results of threat and vulnerability analysis and RM decisions, define prioritized security objectives, if they are not already available;
5. Develop prioritized security and security-related requirements based on the security objectives, taking into account resource constraints, if they are not already available;
6. Carry out System Architecture Decomposition (SAD), focusing on security-relevant parts of the system, and

associate trust values to the components where applicable. This step overlaps with the Service Modeling Step of the BUGYO security assurance methodology;

7. Carry out Security and Security-relevant Requirement Decomposition (SRD) and associate trust values to the components where applicable;
8. Identify the dependencies between SAD and SRD and append the SAD with elements from SRD;
9. Identify candidate Basic Measurable Components (BMCs) from the appended SAD. BMCs are leaf components of the decomposition that clearly manifest a measurable property of the system. They are an abstract way to discuss certain metrics;
10. Integrate security metrics from other sources and select BMCs to be processed to detailed metrics from the candidate BMCs based on a feasibility analysis; and
11. Develop a balanced collection of security metrics from the BMCs to be utilized by the architectural risk analysis. Some metrics can also be utilized in security correctness assurance. In this case, a mapping from these BMCs to different confidence metrics of BUGYO ALs should be developed.

Compared to [1], system architecture decomposition and utilization of trust values have been added here, and the prioritization has been moved to an earlier phase. In addition, the term *security-related requirements* is used instead of *security requirements*.

The first step in the methodology is a threat and vulnerability analysis, with the goal of identifying security threats and their sources and analyzing their likelihood. It is also the starting point of security metrics development, unless sufficient threat and vulnerability information exists beforehand [1]. If architectural risk analysis is carried out continuously within the operational security assurance cycle, updated threat or vulnerability information can potentially trigger new metrics development activity. In measurement architecture, measurable information is identified and the mechanisms for obtaining and processing that data are developed. This step should be carried out together with the actual metrics development. Taxonomies can be used to guide the metrics development; they act as the ‘glue’ between the requirements and the actual design of the SuI. Close attention is required, given that different, even conflicting, taxonomies may be available.

Prioritization of security objectives is mainly based on the RM decisions. Prioritization of security requirements should be aligned with the objective prioritization. However, in addition, resource constraints are taken into account at a more detailed level in the requirements. Security assurance activities focus on certain components of the SuI and the system as a whole. Requirements engineering normally ensures that security requirements are elicited and decomposed to different components and sub-components of the system. However, during the R&D process, due to different constraints, the requirements of components or sub-components might evolve in a different direction from the original requirements that are aligned with the RM decisions. Moreover, due to changed risks during the operation, the system that had satisfied the requirements in the past might not do so anymore. Therefore, it is important to decompose both the actual system and the highest-level valid requirements, and identify the dependencies.

The security requirement decomposition is carried out as follows, based on [16]: (*i*) successive components from each security-related requirement that *contribute to the correctness, effectiveness and/or efficiency* of the requirement are identified, depending on the metrics dimension(s) addressed; (*ii*) the subordinate nodes are examined to determine whether further decomposition is needed. If so, the process is repeated with the subordinate nodes as current goals, broken down to their essential components; and (iii) the decomposition process is terminated when none of the leaf nodes can be decomposed any further, or further analysis of these components is no longer necessary (in other words, once the BMCs have been reached).

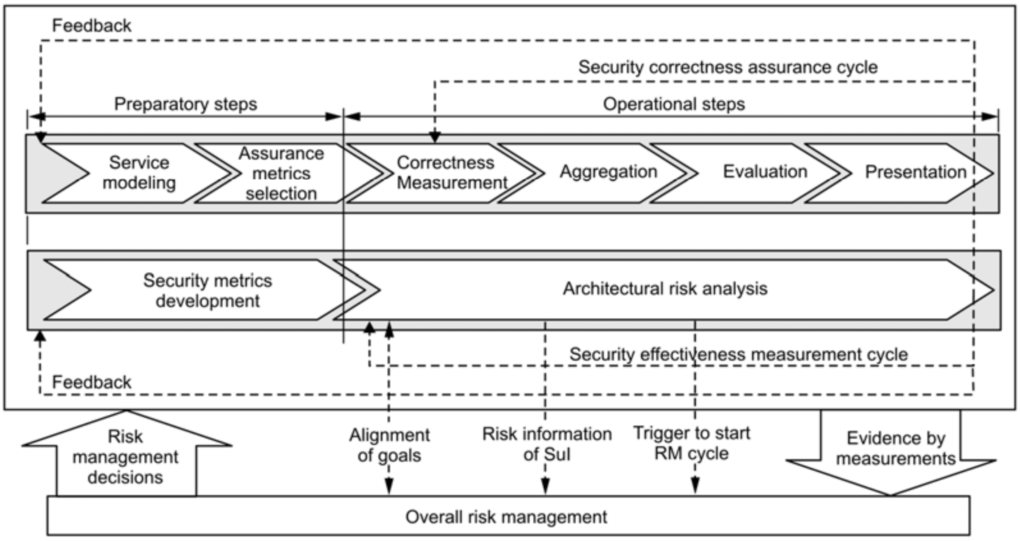

Figure 3. BUGYO security assurance methodology enhanced with security effectiveness measurement.

B. Enhancing BUGYO Security Assurance Methodology

The BUGYO Project, the precursor to BUGYO Beyond, has defined a security assurance methodology for service-oriented systems [13]. The methodology consists of six steps, classified into preparatory and operational steps. During the preparatory steps, the modeled service and the relevant metrics are selected; during the operational steps, measures are gathered, aggregated, evaluated and presented. Table II lists the different Assurance Levels (AL) of the BUGYO approach [13], inspired by the Common Criteria. An increasing AL rating reflects an increasing degree of confidence in the correctness of security controls.

This paper proposes the addition of the following parallel components to the original BUGYO methodology [13] in order to enable measurement of security effectiveness:

1. Security metrics development as parallel activity,
2. Architectural Risk Analysis utilizing security metrics,
3. Interaction with the overall RM to increase the effectiveness of security controls and, consequently, enable measurement of the effectiveness of security controls, and
4. Connection of Service Modeling Step of BUGYO security assurance methodology with Step 6 of the metrics development methodology.

Figure 3 shows phases of a modified BUGYO methodology. Security assurance levels address the confidence in the correctness of security controls, while security metrics address the effectiveness of security controls. The dashed lines indicate the directions of iterative information flows.

TABLE II. BUGYO SECURITY ASSURANCE LEVELS

Assurance Level	Explanation
AL1	Rudimentary evidence for parts
AL2	Regular informal evidence for selected parts
AL3	Frequent informal evidence for selected parts
AL4	Continuous informal evidence for significant parts
AL5	Continuous semi-formal evidence for the entire system

IV. EXAMPLE: PUSH E-MAIL SYSTEM

E-mail is, by nature, an end-to-end system; in other words, its flow goes from Sender to Receiver. The functionality of Push E-mail takes place at the last hop, from the Receiver's E-mail Server to the Receiver's Client, which can reside in a smart phone, for example. Smart phones have become an integral part of everyday life for many people, both in business and leisure. Businesses typically have more stringent requirements for e-mail security in smart phones than spare-time users do. Availability is sometimes the most important requirement, but confidentiality is of primary importance for others. Assume that Alice would like to send an e-mail message to Bob and she knows that Bob's e-mail address is bob@bigcompany.com. The basic sequence of traditional e-mail transfer consists of the following steps.

1. Alice sends an e-mail from her computer (or to be exact from an E-mail Client called Mail User Agent (MUA)) to a Mail Transfer Agent (MTA) in the E-mail Server run by Alice's Internet Service Provider (ISP). From an E-mail Service security assurance point of view, the MUA is an unmanaged object;
2. The MTA requests the address of 'bigcompany.com' from the Domain Name System (DNS). DNS service is also an unmanaged object;
3. The DNS responds with the address information;
4. Alice's MTA sends the message to Bob's MTA using Simple Mail Transfer Protocol (SMTP); and
5. Bob's MTA send the message to his MUA using the Post Office Protocol version 3 (POP3) or the Internet Message Access Protocol (IMAP) protocol.

If the e-mail address is on the local server, the message is passed to the Mail Delivery Agent (MDA) of the server instead of the MTA. In general, the e-mail system is a push system, but the last-hop protocols POP3 and IMAP provide polling, which is used in the last hop because the E-mail Server does not always know the actual address of the E-mail Client. The clear demand for 'always-on' capability in e-mail systems, especially in mobile phones and other hand-held devices, has resulted in Push E-mail systems. The current implementations consist of two different approaches: Notification Push E-mail and True Push E-mail. With the former, the client asks to be notified of any change in the E-mail Server, such as a new incoming e-mail message. The IMAP provides functionality for the notification. The use of IMAP requires that the e-mail client has a TCP/IP connection to the e-mail server. In True Push E-mail, the server simply pushes the entire incoming e-mail to the client when it arrives in the server.

A. Security Threats and SecurityRequirements

Table III presents some threats to the example service in terms of confidentiality, integrity, and availability, while Table IV lists some security requirements. For the sake of simplicity, security-related requirements that are not security requirements (mostly software quality requirements) are not covered here.

TABLE III. SOME EXAMPLES OF THREATS TO THE EXAMPLE SYSTEM

#	Type	Threat
T1	Confidentiality	Attacker has access to the system as an administrator or e-mail user.
T2	Integrity	The content or header of an e-mail message is modified in the E-mail Server.
T3	Availability	A DoS or DDoS attack causes delay to e-mail service or the server crashes.

The main security threats to the Push-Email system include Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. If authentication and access control fail, an attacker can access the e-mail server and even seize the system. For the sake of simplicity, this paper does not address additional security threats such as eavesdropping, masquerading attacks, corruption of content, and data processing functionality, spam, and phishing. Although it is important, investigation of vulnerability management is

978-1-4244-5494-5/10/$26.00 ©2010 IEEE

omitted for the same reason. Like threats, vulnerabilities of the SuI should be identified and managed with suitable requirements.

TABLE IV. SOME EXAMPLES OF SECURITY REQUIREMENTS

#	T	Function	Requirement
R1	T1	User authentication	Any user accessing the e-mail client should be authenticated and the client should be aware of the authentication strength.
R2	T1	E-mail server authentication	The sender's e-mail server should be authenticated to the DNS (source authentication)
R3	T1	Identity management	User identities should be securely managed.
R4	T1	Access control mechanism	Access control can use different access control mechanisms depending on the authentication level.
R5	T1	User authorization	The Authorization Manager should verify user identity and grant access to the resource allowed.
R6	T1	Revoking authorization	The authorization functionality should support the revocation of authorization, for example, to users identified as harmful.
R7	T2	Spam and malicious attachments	The system should remove spam messages and malicious attachments from messages and notify the user of removals.

B. Development of Measurement Architecture

Appropriate mechanisms to obtain the necessary measurements from the SuI must be developed 'hand-in-hand' with the metrics. Built-in security-measurability-enhancing mechanisms [17] enable effective and efficient use of metrics.

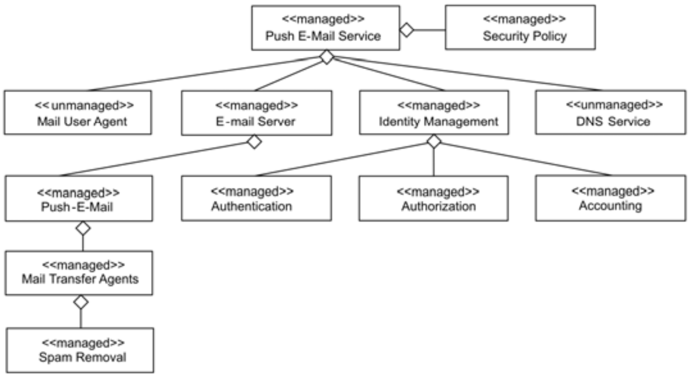
<<managed>> Push E-Mail Service
<<managed>> Security Policy
<<unmanaged>> Mail User Agent
<<managed>> E-mail Server
<<managed>> Identity Management
<<unmanaged>> DNS Service
<<managed>> Push-E-Mail
<<managed>> Authentication
<<managed>> Authorization
<<managed>> Accounting
<<managed>> Mail Transfer Agents
<<managed>> Spam Removal

Figure 4. Some components of the Push E-mail service.

C. System Architecture and Requirement Decomposition

Figure 4 shows a partial service architecture diagram of an example Push E-mail service, emphasizing the *system decomposition* that leads to authentication and access control functions (related to requirements R1, R2, R3, R4, R5, R6) and spam filtering (R7). The Mail User Agent resides on the user's computer and is therefore labeled as 'unmanaged'. The DNS service is also assumed to be an external 'unmanaged' service from the perspective of the e-mail service provider. Functions other than Identity Management of the E-mail server have been omitted from the diagram. The spam filtering, including removal of malicious message attachments, is carried out by the service. Another possibility is to leave the responsibility to the end-user. In the decomposition, trust values are associated with the unmanaged components and managed components with no assurance evidence. For example, the trust value of the DNS Service can be 0.9 and that of a particular MUA can be 0.3. After the security *requirements* have been defined, they are decomposed in order to be able to identify the Requirement BMCs and parameter dependencies. Pre-existing taxonomies and classifications can be utilized, as shown below. If several taxonomies exist for a special security objective, validation techniques for qualitative information, such as triangulation, can be used to construct a suitable taxonomy. Figure 5 shows an example of generic authentication decomposition, based on [16]. The generic authentication decomposition can be applied to Requirement R1 (user authentication) and R2 (E-mail Server authentication) of Table III, resulting in specific decompositions for R1 and R2. The authentication strength required by R1 can be based on an aggregated metric of BMCs. This decomposition does not address identity management in general (R3).

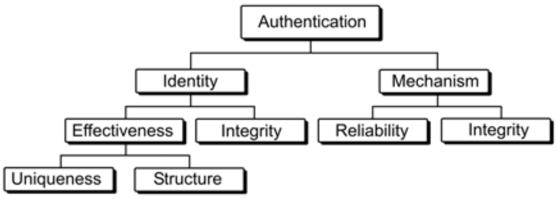

Figure 5. Example of generic authentication requirement decomposition [16].

The U.S. National Institute of Standards and Technology (NIST) document, the Electronic Authentication Guideline [18], classifies the Level of Assurance (LoA in NIST terminology) according to four different levels ranging from 1 to 4. This guideline applies a weakest-link approach: LoA is the lowest level, which is reached from five confidence metrics: registration and issuance, tokens, token and credential management, authentication process and assertions. For example, Table V summarizes the requirements for tokens confidence metric. Similar requirements have been defined for each confidence metric area. As the LoA classification is defined as be a generic assurance system for electronic authentication, specific security requirements existing in the particular SuI might require additional processing and decomposition. Although both LoA (1…4) and BUGYO AL (1…5) are both assurance metrics, the difference is that LoA addresses security effectiveness, while AL deals with security correctness. It is simple, therefore, to create a mapping between these two classifications: ALs can be used to measure confidence of security correctness with each LoA; in fact, ALs can be understood as 'subclasses' of LoAs. Note that the weakest-link assessment technique of NIST's LoA system is different from BUGYO's approach of weighing confidence.

Trust values should be associated to the node of the decomposition where applicable. For example, a trust value can be attached to the identity branch of Figure 5 if no identity uniqueness, structure and integrity information is available. In this case, the authentication strength would consist of the reliability and integrity of the authentication mechanism and the trust value for the identity branch. When the DNS was originally developed, the design focused on availability. Consequently, just one malicious (false) server could disrupt the entire DNS, raising the need for source authentication. Because DNS is an unmanaged system from the point of view of this Push E-mail service, a trust value needs to be associated with the DNS source authentication.

TABLE V. REQUIREMENTS FOR TOKENS CONFIDENCE METRIC OF NIST AUTHENTICATION LOA [18]

LoA	Requirements
1	Password, pre-registered knowledge, look-up secret, out of band
2	Password, pre-registered knowledge, look-up secret, out of band, single factor one-time password device, single factor cryptographic device
3	Multi-factor software cryptographic token
4	Multi-factor one-time password hardware token, multifactor hardware cryptographic token

D. Dependencies

The identity management and authentication components of service decomposition in Figure 4 clearly have dependencies in the authentication requirement decomposition shown in Figure 5, or, alternatively, the NIST LoA decomposition. Moreover, authorization depends on the authentication and access control (see Figure 6) [1]. This decomposition is applied to requirements R4, R5, and R6, resulting in specific decompositions.

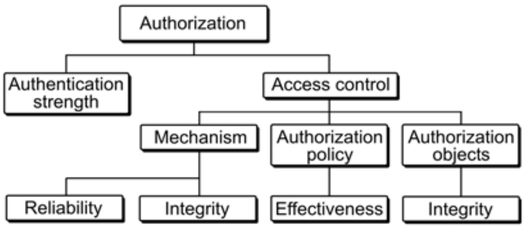

Figure 6. Example of generic authorization requirement decomposition [1].

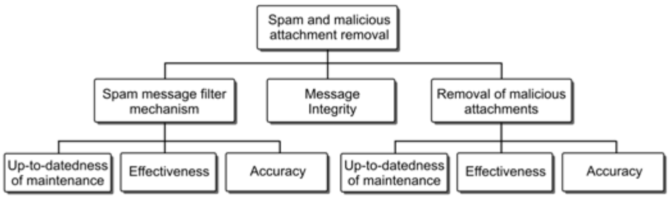
Spam and malicious attachment removal
Spam message filter mechanism
Message Integrity
Removal of malicious attachments
Up-to-datedness of maintenance
Effectiveness
Accuracy
Up-to-datedness of maintenance
Effectiveness
Accuracy

Figure 7. A spam and malicious attachment removal decomposition.

The authorization policy of Figure 6 and the security policy of Figure 3 clearly have dependencies as well. Decomposing requirement R7 (spam and malicious attachments) yields to the example decomposition of Figure 7. The dependencies should be modeled appropriately; for instance, by adding new components from the requirement decomposition to the system architecture decomposition diagrams. This approach is also effective because the candidate BMCs are allocated to the adequate system components, enabling the feasibility analysis to yield to the selection of the final BMCs.

E. Identification of Basic Measurable Components

Table V shows the BMCs from Figure 6. The E-mail Metrics Program of the Messaging Anti-Abuse Working Group (MAAWG) [19] has defined some spam filter effectiveness metrics for ISPs, which are presented in Table VI. *S/D* can be utilized to represent *SFE* of Table VII. Development of *SFA* and *MAA* accuracy metrics can be based on the ratio of false positives (legitimate messages that are misidentified by the mechanisms) and false negatives (spam messages or malicious attachments).

TABLE VI. BMCS FOR SPAM AND MALICIOUS ATTACHMENT REMOVAL

Symbol	Basic Measurable Component
SFU	Spam Message Filter Mechanism Up-to-Datedness
SFE	Spam Message Filter Effectiveness
SFA	Spam Message Filter Accuracy
MAU	Malicious Attachment Removal Up-to-Datedness
MAE	Malicious Attachment Removal Effectiveness
MAA	Malicious Attachment Removal Accuracy

TABLE VII. METRICS FOR E-MAIL SPAM FILTER EFFECTIVENESS

Symbol	Metrics
S	Number of dropped connections and blocked or tagged inbound e-mails per mailbox
S/D	Ratio of dropped connections and blocked or tagged inbound e-mails to unaltered delivered email
D	Number of unaltered delivered e-mail per mailbox

F. Integration of Metrics from Different Sources and Balanced Collection of them

Metrics from sources other than requirement and system architecture decomposition can be integrated into the set of metrics, based on an analysis of how they are related to the requirements. For example, it is possibly to measure the compliance of the Push E-mail service with the privacy, non-repudiation and person register regulations and legislation. Moreover, compliance with appropriate standards and Requests for Comments (RFCs) is also measurable. Other technical or quality parameters (e.g., Quality of Service) that are available in the SuI can often be utilized as a part of security metrics. Earlier work by the authors of this paper investigated the feasibility criteria for utilizing security metrics in software-intensive systems [20]. These criteria can be utilized for the selection of metrics. In order to aggregate component metrics into summed metrics, different weights can be associated with different component metrics, indicating the relative importance among the component metrics. A 'close to correct' weight assignment is used in practice because there are no analytical results for determining the relative priorities of the elements, other than the careful use of one's expertise and judgment [15]. Certain security metrics can also be used in security correctness measurement. In this case, a mapping from them should be developed to the component confidence metrics of ALs.

V. Related Work

Despite several major attempts to standardize security evaluation and certification of technical systems, they have only achieved limited success in terms of advancing security measurability [6]. This is largely because the standards are rigid and created for certification and carrying out these processes requires significant amounts of time and money. The most widely used of these efforts is the Common Criteria (CC) ISO/IEC 15408 International Standard [4] for security evaluation by human evaluators which focuses primarily on documentation rather than the actual security of the operational system. During the CC evaluation process, an Evaluation Assurance Level (EAL), from EAL1 to EAL7, is assigned to the SuI to describe the depth and rigor of an evaluation. Most CC-evaluated products to date have been information system components, such as firewalls and smart cards. Related work has been undertaken towards security measurability, especially in the area of software development, such as in structured assurance [21]. However, it is not possible to achieve complete identification of security threats during system development. Even if it was possible, assurance on security controls is very difficult once they have been deployed. Some researchers, such as Pham et al. [22], have suggested using attack graphs and anomaly detection metrics. However, this approach lacks security effectiveness metrics. Penetration testing [23] is another commonly used technique for evaluating the security of computer systems or networks by simulating an attack from a malicious source. However, penetration testing remains an on/off system audit that is generally performed prior to system deployment. Surveys of security metrics approaches can be found in [20], [24], [25], [26], [27], and [28].

VI. Conclusions and Future Work

Although systematic approaches to quantification of security in order to support security assurance activities are desirable, they are also rare: widely accepted taxonomies, models, methodologies, and tools are missing. This paper has introduced a novel methodology to implement measurement of security effectiveness in connection with the security control correctness assurance of service-centric systems. The methodology is risk-driven and utilizes the decomposition of both security requirements and system architecture as a basic mechanism. The decompositions make it possible to identify Basic Measurable Components yielding to actual security metrics. Our future work includes defining of component confidence metrics for Security Assurance Levels, developing confidence metrics for security effectiveness, and investigating aggregation and taxonomy selection techniques for security metrics for security assurance.

References

[1] R. Savola and H. Abie, “Development of measurable security for a distributed messaging system,” International Journal of Advances in Security, Vol. 2, No. 4, 2010.

[2] H. Abie, I. Dattani, M. Novkovic, J. Bigham, S. Topham and R. Savola, “GEMOM – Significant and measurable progress beyond the state of the art,” ICSNC ’08, 26–31 Oct. 2008.

[3] BUGYO Beyond Project. Website available: www.celtic-initiative.org/Projects/BUGYO-BEYOND/default.asp [June 29, 2010].

[4] ISO/IEC 15408-1:2005, “Common Criteria for information technology security evaluation – Part 1: Introduction and general model,” ISO/IEC, 2005.

[5] R. Savola, “A security metrics taxonomization model for software-intensive systems,” Journal of Information Processing Systems, Vol. 5, No. 4, Dec. 2009, pp. 197–206.

[6] W. Jansen, “Directions in security metrics research,” U.S. National Institute of Standards and Technology, NISTIR 7564, Apr. 2009, 21 p.

[7] R. Savola, “A taxonomical approach for information security metrics development,” NORDSEC ’07, 2007.

[8] Information Technology Security Evaluation Criteria (ITSEC), Version 1.2, Commission for the European Communities, 1991.

[9] ISO/IEC 27000:2009: Information technology – Security techniques – Information security management systems – Overview and vocabulary. ISO/IEC, 2009.

[10] A. Zuccato, B. Marquet, S. Papillon, M. Aldén, “Service oriented modeling of communication infrastructure for assurance,” IAW ‘06, United States Military Academy, West Point, N.Y., 2006.

[11] M. Ouedraogo, D. Khadraoui, B. de Rémont, E. Dubois, H. Mouratidis, “Deployment of a security assurance monitoring framework for telecommunication service infrastructure on a VoIP system,” NTMS ’08, 5–7 Nov. 2008.

[12] R.M. Kanter, “Confidence: leadership and the psychology of turnarounds,” Random House, London, 2004.

[13] A. Zuccato, S. Dubus, E. Bulut, “Methodology for service-oriented management of security assurance in communication infrastructures,” HASE ’08, 2008, pp. 483–486.

[14] G. McGraw, “Software security – building security in,” Addison-Wesley, 2006.

[15] M. Howard and D. LeBlanc, “Writing secure code,” Microsoft, 2003.

[16] C. Wang and W.A. Wulf, “Towards a framework for security measurement,” NISSC ’97, 1997, pp. 522–533.

[17] R. Savola and P. Heinonen, “Security-measurability-enhancing mechanisms for a distributed adaptive security monitoring system,” SECURWARE ’10, 2010, 10 p.

[18] W.E. Burr et al., “Electronic authentication guideline,” National Institute of Standards and Technology, U.S. Department of Commerce, NIST Special Publication 800-63-1, Draft, Dec. 8, 2008, 97 p.

[19] Messaging Anti-Abuse Working Group (MAAWG): www.maawg.org [June 29, 2010].

[20] R. Savola, “On the feasibility of utilizing security metrics in software-intensive systems,” International Journal of Computer Science and Network Security, Vol. 10, No. 1, Jan. 2010, pp. 230–239.

[21] T.S. Ankrum, A.H. Kromholz, “Structured assurance cases: three common standards,” HASE ’05, 2005, pp. 99–108.

[22] N. Pham, L. Baud, P. Bellot and M. Riguidel, “A near real-time system for security assurance assessment,” ICIMP ’08, 2008.

[23] T.J. Klevinsky, S. Laliberte, A. Gupta, “Hack I.T.: security through penetration testing,” Addison-Wesley, Boston, MA, USA, 2002.

[24] D. S. Herrmann, “Complete guide to security and privacy metrics – measuring regulatory compliance, operational resilience and ROI,” Auerbach Publications, 2007, 824 p.

[25] A. Jaquith, “Security metrics: replacing fear, uncertainty and doubt,” Addison-Wesley, 2007.

[26] N. Bartol, B. Bates, K.M. Goertzel, and T. Winograd, “Measuring cyber security and information assurance: a state-of-the-art report,” Information Assurance Technology Analysis Center IATAC, May 2009.

[27] V. Verendel, “Quantified security is a weak hypothesis: a critical survey of results and assumptions,” New Security Paradigms Workshop, Oxford, U.K., 2009, pp. 37–50.

[28] R. Savola, “A novel security metrics taxonomy for R&D organisations,” ISSA ‘08, 2008, pp. 379–390.

978-1-4244-5494-5/10/$26.00 ©2010 IEEE