First International Workshop on Measurability of Security in Software Architectures – MeSSa 2010

Reijo M. Savola
VTT Technical Research Centre of Finland
Kaitoväylä 1, P.O. Box 1100
90571 OULU, Finland
+358 40 569 6380
Reijo.Savola@vtt.fi

Teemu Kanstrén
VTT Technical Research Centre of Finland
Kaitoväylä 1, P.O. Box 1100
90571 OULU, Finland
+358 40 548 5746
Teemu.Kanstren@vtt.fi

Antti Evesti
VTT Technical Research Centre of Finland
Kaitoväylä 1, P.O. Box 1100
90571 OULU, Finland
+358 40 552 7542
Antti.Evesti@vtt.fi

ABSTRACT

The growing complexity of service-centric systems has increased the need for pertinent and reliable software security and trusted system solutions. Systematic approaches to measuring security in software architectures are needed in order to obtain sufficient and credible proactive evidence of the security level or performance of a system, service or product. The systematic definition of security metrics and security assurance metrics is a young field that still lacks widely accepted definitions of metrics and applicable measuring techniques for design-time and run-time security monitoring. MeSSa 2010 workshop contributes on the following issues:

- Security, trust and privacy metrics
- Security assurance metrics
- Security, trust and privacy measurement systems and associated data gathering
- Metrics for adaptive security systems
- Taxonomical and ontological research on security metrics
- Experimental results from security measurements
- Security measurability-increasing mechanisms for software architectures
- The relationship and differences between security metrics and security assurance metrics
- Trade-off analysis and decision-making at design-time and at run-time.

General Terms

Algorithms, Management, Measurement, Performance, Design, Economics, Reliability, Experimentation, Security, Standardization, Theory, Verification.

Keywords

Security, privacy, trust, measurement, metrics, assessment, evaluation, assurance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ECSA 2010, August 23–26, 2010, Copenhagen, Denmark.
Copyright (c) 2010 ACM 978-1-4503-0179-4/10/08 …$10.00.

1. ORGANIZERS

MeSSa 2010 is co-located with the 4th European Conference on Software Architecture (ECSA 2010).

1.1 Organizing Projects

MeSSa 2010 is co-organized by the following European research projects: GEMOM[1] (Genetic Message Oriented Secure Middleware), BUGYO Beyond[2] (Building Security Assurance in Open Infrastructures, Beyond), and SOFIA[3] (Smart Objects For Intelligent Applications).

GEMOM (Genetic Message Oriented Secure Middleware) is an EU FP7 ICT project (2008–2010) that focuses on significant and measurable increases in the end-to-end intelligence, security and resilience of complex distributed information systems. The GEMOM project has prototyped a security monitoring system that utilizes security metrics and has developed novel approaches for security metrics development. The prototypes are currently being validated in five case studies – a collaborative business portal, a dynamic linked exchange, a financial market data delivery system, a dynamic road management system and a banking scenario.

BUGYO Beyond (Building Security Assurance in Open Infrastructures, Beyond) is a CELTIC Eureka project (2008–2011) that focuses on extending the security assurance metrics work performed in the BUGYO CELTIC project. Its aims are: (i) to cover areas such as self-developed metrics, patterns of metrics and modelling support, (ii) to provide means for comparing and exchanging assurance information between different operators, including the normalization and standardization of assurance measures and aggregated levels, and (iii) to cope with dynamics and mobility by addressing issues that emerge from evolving and ubiquitous infrastructure.

SOFIA (Smart Objects For Intelligent Applications) is an ARTEMIS project (2009–2011) that focuses on the information interoperability of physical spaces and the ontology-driven development of smart space applications. One of the key drivers in the development of smart spaces and smart space applications is information security and its run-time management in changing

[1] www.gemom.eu

[2] www.celtic-initiative.org/Projects/BUGYO-BEYOND/default.asp

[3] www.sofia-project.eu

situations. The first prototyped solutions of the run-time security management suggest that common and widely accepted security metrics ontology could be developed as a joint effort between European research projects.

1.2 Workshop Co-Chairs

Reijo Savola, VTT Technical Research Centre of Finland

Teemu Kanstrén, VTT Technical Research Centre of Finland

Antti Evesti, VTT Technical Research Centre of Finland

1.3 Technical Program Committee

Habtamu Abie, Norwegian Computing Center (Norway)

Nadya Bartol, Booz Allen Hamilton (USA)

Ulrike Baumann, EADS (France)

John Bigham, Queen Mary University of London (UK)

Christophe Blad, Oppida (France)

Jim Clarke, Waterford Institute of Technology (Ireland)

Marijke Coetzee, University of Johannesburg (South Africa)

Michel Cukier, University of Maryland (USA)

Giorgio da Bormida, ELGI (Italy)

Ilesh Dattani, Q-Sphere (UK)

Samuel Dubus, Alcatel-Lucent (France)

Sammy Haddad, ENST (France)

Perttu Halonen, Nokia Siemens Networks (Finland)

Artur Hecker, Telecom ParisTech (France)

Thomas Heyman, KU Leuven (Belgium)

Zoltan Hornák, SEARCH-LAB (Hungary)

Siv-Hilde Houmb, Telenor (Norway)

Erland Jonsson, Chalmers University of Technology (Sweden)

Oscar López, Nextel S.A. (Spain)

Louis Marinos, ENISA (Greece)

Aliki Ott, Nokia Siemens Networks (Finland)

Moussa Ouedraogo, CRP Henri Tudor (Luxembourg)

Eila Ovaska, VTT, Finland

Tanir Ozcelebi, TU Eindhoven (the Netherlands)

Pierre Parrend, Karlsruhe Institute of Technology (Germany)

Aljosa Pasic, Atos Origin (Spain)

Christophe Ponchen, EADS (France)

Michel Riguidel, ENST (France)

Domenico Rotondi, TXT e-solutions S.p.A. (Italy)

Juha Röning, University of Oulu (Finland)

Riccardo Scandariato, KU Leuven (Belgium)

Pedro Soria-Rodriguez, Atos Origin (Spain)

Ari Takanen, Codenomicon (Finland)

Alessandra Toninelli, INRIA Paris (France)

Hein Venter, University of Pretoria (South Africa)

Antti Vähä-Sipilä, Nokia (Finland)

2. ACCEPTED PAPERS

The following papers were accepted to the workshop, listed here in alphabetical order according to the first author’s surname.

2.1 Peer-Reviewed Papers

[1] **Indicator-based Architecture-level Security Evaluation in a Service-oriented Environment.** Pablo Antonino, Slawomir Duszynski, Christian Jung and Manuel Rudolph.

A method called SiSOA for security evaluation of existing complex service-oriented systems at architectural level is introduced. The method is based on reverse engineering techniques and utilizes a knowledge base.

[2] **Applicability of Security Metrics for Adaptive Security Management in a Universal Banking Hub System.** Lorenzo Blasi, Reijo M. Savola, Habtamu Abie and Domenico Rotondi.

Experiences from deployment of security metrics-driven adaptive security solutions for a distributed message-oriented middleware are discussed. The metrics are developed utilizing a risk-driven approach described in the authors’ earlier work.

[3] **Towards Micro Architecture for Security Adaptation**. Antti Evesti and Susanna Pantsar-Syväniemi.

A micro-architecture for security adaptation and associated context information taxonomy for smart spaces are introduced. The micro-architecture contains six execution phases, one of which being context monitoring.

[4] **Security Measurements within the Framework of Quality Assessment Models for Free/Libre Open Source Software.** Arne-Kristian Groven, Kirsten Haaland, Ruediger Glott and Anna Tannenberg.

Two quality assessment models, OpenBRR and QualOSS, are compared in the context of a telephone private branch exchange case study. Various aspects of the approaches, including security measurement capabilities, are addressed.

[5] **Towards Holistic Security Management through Coherent Measuring.** Perttu Halonen and Kimmo Hätönen.

Some technical problems and the big picture of security management in the context of complex communication systems are discussed. The paper proposes coherent measurement of various technical aspects of security and utilization of security impact metrics.

[6] **Formal Approach to Security Metrics -- What does "More Secure" mean for you?** Leanid Krautsevich, Fabio Martinelli and Artsiom Yautsiukhin.

A basic model for formal description and analysis of security metrics is introduced. Dependencies of metrics and attacker *models are also investigated. Furthermore, relation "more security" is discussed.*

[7] **Comparison of Software Design Security Metrics.** Daniel Mellado, Eduardo Fernández-Medina and Mario Piattini.

A few widely-known security design approaches for software products with metrics are discussed and compared. Various capability aspects of the approaches are compared and summarized.

[8] **On the Effectiveness of the Metamorphic Shield.** Anh Nguyen-Tuong, Andrew Wang, Jason D. Hiser, John C. Knight, and Jack W. Davidson.

An artificial diversity security model for metamorphosis of attack surface called Metamorphic Shield is introduced. The model is applied to an incremental attack against instruction set randomization.

[9] **Risk Analysis of Host Identity Protocol -- Using Risk Identification Method Based on Value Chain Dynamics Toolkit.** Juha Sääskilahti and Mikko Särelä.

A risk identification method based on Value Chain Dynamics Toolkit is introduced and applied to risk analysis of Host Identity Protocol. The method offers benefits in knowledge transfer, structuring of interviews and visualization of value chains.

[10] **Trust-terms Ontology for Defining Security Requirements and Metrics.** Kieran Sullivan, Jim Clarke and Barry P. Mulcahy

Trust-terms ontology for various components and concepts that comprise ICT security and trust is proposed. The ontology helps in gaining a better understanding of trust and security requirements and in identifying more precise measurability criteria.

[11] **Secure Information Sharing between Heterogeneous Embedded Devices.** Jani Suomalainen, Pasi Hyttinen and Pentti Tarvainen.

A novel security architecture for smart spaces enabling heterogeneous devices to share data in controlled manner is introduced. Centralized information brokering device is used to measure security level of published information.

2.2 Invited Paper

[12] **Towards an Abstraction Layer for Security Assurance Measurements (Invited Paper).** Teemu Kanstrén, Reijo Savola, Antti Evesti, Heimo Pentikäinen, Artur Hecker, Moussa Ouedrago, Kimmo Hätönen, Perttu Halonen, Christophe Blad, Oscar López and Saioa Ros

An approach for creation of an Abstraction Layer of security assurance measurements from the requirements, and vice versa, is introduced. The approach is discussed in a security assurance case example of Push E-mail service system.

3. WORKSHOP PROGRAM

09:00—09:10 **Welcome by the organizers**

Session 1: Foundations of Security Measurement

09:10—09:30 **Towards Holistic Security Management through Coherent Measuring**

Perttu Halonen and Kimmo Hätönen

09:30—09:50 **Formal Approach to Security Metrics – What does "More Secure" Mean to you?**

Leanid Krautsevich, Fabio Martinelli and Artsiom Yautsiukhin

09:50—10:10 **On the Effectiveness of the Metamorphic Shield**

Anh Nguyen-Tuong, Andrew Wang, Jason D. Hiser, John C. Knight and Jack W. Davidson

Session 2: Taxonomy and Ontology-based Approaches

10:10—10:30 **Trust-terms Ontology for Defining Security Requirements and Metrics**

Kieran Sullivan, Jim Clarke and Barry P. Mulcahy

10:30—10:50 **Towards Micro Architecture for Security Adaptation**

Antti Evesti and Susanna Pantsar-Syväniemi

10:50—11:10 **Towards an Abstraction Layer for Security Assurance Measurements (Invited Paper)**

Teemu Kanstrén, Reijo Savola, Antti Evesti, Heimo Pentikäinen, Artur Hecker, Moussa Ouedraogo, Kimmo Hätönen, Perttu Halonen, Christophe Blad, Oscar López and Saioa Ros

11:10—11:30 **Break**

Session 3: Specific Applications of Security Metrics

11:30—11:50 **Applicability of Security Metrics for Adaptive Security Management in a Universal Banking Hub System**

Lorenzo Blasi, Reijo M. Savola, Habtamu Abie and Domenico Rotondi

11:50—12:10 **Secure Information Sharing between Heterogeneous Embedded Devices**

Jani Suomalainen, Pasi Hyttinen and Pentti Tarvainen

12:10—12:30 **Risk Analysis of Host Identity Protocol – Using Risk Identification Method Based on Value Chain Dynamics Toolkit**

Juha Sääskilahti and Mikko Särelä

12:30—14:00 **Lunch**

Session 4: Security Evaluation and Software Design

14:00—14:20 **Indicator-based Architecture-level Security Evaluation in a Service-oriented Environment**

Pablo Antonino, Slawomir Duszynski, Christian Jung and Manuel Rudolph

14:20—14:40	**Security Measurements within the Framework of Quality Assessment Models for Free/Libre Open Source Software** Arne-Kristian Groven, Kirsten Haaland, Ruediger Glott and Anna Tannenberg
14:40—15:00	**Comparison of Software Design Security Metrics** Daniel Mellado, Eduardo Fernández-Medina and Mario Piattini
15:00—15:10	**Break**
15:10—16:40	**Panel Discussion on Future Research Directions of Security, Privacy and Trust Measurement**
16:40—16:50	**Closing**

4. ADVANCES OF THE WORKSHOP AND CHALLENGES FOR THE FUTURE

Systematic approaches for security, privacy and trust metrics development and deployment are much desired but rare. There are various reasons for this situation, for example: current practice of information security is a highly diverse field, there is lack of measurability-enhancing mechanisms in systems, and availability of meaningful security-relevant evidence is often poor. Widely accepted security metrology models, methods and tools have been missing. New emerging and evolving service, communication and software technologies and paradigms, such as cloud services, increase the needs for sufficient and credible security evidence. It is clear that open discussion and cross-disciplinary research co-operation is needed to make advances in this field.

MeSSa 2010 addresses research and experimentation results in the development of security metrics, security assurance metrics and security measurement solutions in the context of software-intensive systems, particularly on the software architecture level of service-centric systems. The workshop brings experts from security engineering, security assurance, security management, risk analysis, telecommunications engineering and software engineering together to discuss the above mentioned topics and to find answers to the current challenges. GEMOM, BUGYO Beyond and SOFIA project, and lots of other projects and efforts active in the field disseminate their results and stimulate discussions about further research during the workshop.

The Panel Discussion Session of the MeSSa 2010 workshop discusses and identifies potential and meaningful future research directions of security, privacy and trust measurement, paving the path for innovation of more systematic, more practical and more widely used approaches.

A Comparison of Software Design Security Metrics

Daniel Mellado
University of Castilla-La Mancha. GSyA Research Group, Information Systems and Technologies Department
Cobertizo de San Pedro Mártir, 45071 Toledo, Spain
damefe@esdebian.org

Eduardo Fernández-Medina, Mario Piattini
University of Castilla-La Mancha; Information Systems and Technologies Department. Institute of Information Technologies & Systems
Paseo de la Universidad 4, 13071 Ciudad Real, Spain
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

ABSTRACT

A lack of security metrics signifies that it is not possible to measure the success of security policies, mechanisms and implementations, and security cannot, in turn, be improved if it cannot be measured. The importance of the use of metrics to obtain security quality is thus widely accepted. However, the definition of security metrics concerns a discipline which is still in its first stages of development, meaning that few documented resources or works centring on this subject exist to date. In this paper we shall therefore study the latest existing models with which to define security metrics and their components as aspects that have a bearing on the quality of software products with the intention that this will serve as a basis for continued advancement in research into this area of knowledge.

Keywords

Security, metrics, measures, security metrics, design.

1. INTRODUCTION

The current tendency towards information systems which are increasingly bigger and are distributed throughout the entire Internet has led to the emergence of many new threats to security [21]. This signifies that present-day information systems are vulnerable to a host of threats and cyber-attacks by cyber-terrorists, hackers, etc., such as virus which are propagated through the Internet, social engineering attacks (*phishing* etc.) or the inappropriate use of the net's assets by companies' employees [4].

The security in computing has in fact grown tremendously since the 1970s, leading to a huge number of techniques, models, protocols etc. These have also been accompanied by a significantly noticeable amount of activity by international organisations with regard to standardisation and certification. This has taken place to such a great extent that, as is indicated in [13], it is possible to find numerous international standardization organizations that have created a complex structure of standards regarding themes related to information security, which are frequently altered and updated.

It is widely accepted that metrics are important in information security, since without security metrics it is not possible to measure the success of security policies, mechanisms and implementations, and security cannot be improved if it cannot be measured. They can therefore be considered as an effective tool which allows IT security experts to measure and evaluate the strength of security and the levels of its systems, products, processes and preparation in order to manage the security aspects in which they are imbued. Metrics can also assist in the identification of a system's vulnerabilities, thus providing a guide towards the amount of priority that should be given to corrective actions, and raising the level of consciousness with regard to security within the organisation [26]. This even signifies that various laws and standards cite security metrics as requirements as occurs, for example, in the United States with the "Federal Information Security Management Act" or the "Clinger-Cohen Act" among others.

Given the importance of the use of metrics in the quality of the security, the majority of the quality or non-functional requirements have been extensively studied and measured. With regard to security attributes, metrics have even been defined which permit security to be evaluated at the system level and at the code implementation level. Various standards concerning security metrics have been published, such as the Common Criteria [11], ISO/IEC 27004 [12], NIST 800-55 [25] or o FIPS 140-1/2 [6]. However, these regulations and standards are broad and provide imprecise definitions of security metrics, or are too limited to cover a wide variety of security situations [26]. Nevertheless, security is difficult to measure, which causes high instability in security metrics, since the measurement of security, i.e., the definition of security metrics, concerns a discipline which is in its first stages of development and about which there are few documented resources or works [27].

The objective of this paper is, therefore, to analyse the existing models that define security metrics and their components as aspects that have a bearing on the quality of software products.

The remainder of this paper has therefore been organised as follows. Section 2 will present some of the most relevant software design security metrics proposals. The Section 3 will then go on to analyse all the security metrics proposed by the works studied in the previous section from a comparative point of view. Finally, our main conclusions will be presented in Section 4.

2. RELEVANT SECURITY METRICS

This section describes the most important aspects of the design security metrics proposals defined in the principal and most widely accepted standards with regard to security and software

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ECSA 2010, August 23–26, 2010, Copenhagen, Denmark.
Copyright (c) 2010 ACM 978-1-4503-0179-4/10/08 …$10.00.

quality, along with the most outstanding approaches or frameworks.

2.1 Security metrics for object-oriented class diagrams [1]

This approach centres on the security in the design of object oriented applications, such that it defines a series of metrics for this type of applications. These metrics allow designers to discover and resolve security vulnerabilities during the early stages of software development, and help to compare the security in the various design alternatives. The authors specifically propose seven security metrics with which to measure the encapsulation (accessibility) and cohesion (interaction) of data for a determined class from the perspective of potential information losses. The proposed metrics are the following: Classified Instance Data Accessibility (CIDA); Classified Class Data Accessibility (CCDA); Classified Operation Accessibility (COA); Classified Mutator Attribute Interactions (CMAI); Classified Accessor Attribute Interactions (CAAI); Classified Attributes Interaction Weight (CAIW); Classified Methods Weight (CMW).

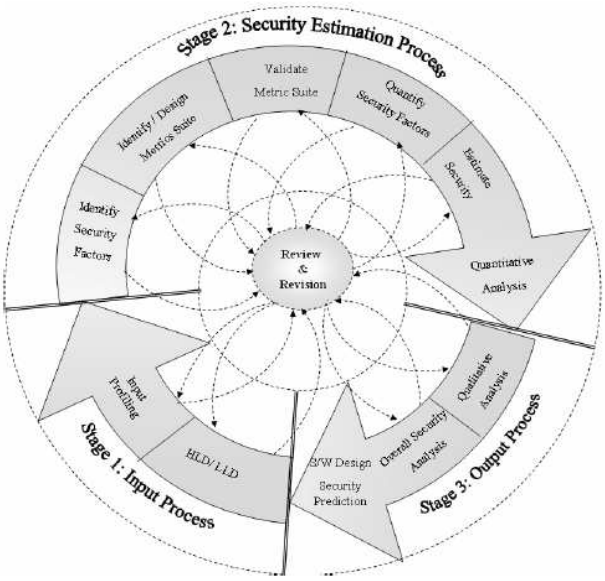

Stage 2: Security Estimation Process
Validate Metric Suite
Quantify Security Factors
Estimate Security
Quantitative Analysis
Identify / Design Metrics Suite
Identify Security Factors
Review & Revision
Input Profiling
HLD/LLD
Stage 1: Input Process
S/W Design Security Prediction
Overall Security Analysis
Qualitative Analysis
Stage 3: Output Process

Fig. 1 – Lifecycle of security estimation

2.2 Security estimation framework: design phase perspective [3]

The authors of this work propose a framework with which to estimate software security from the first phases of the software development lifecycle, in such a way that this framework will allow security experts to estimate software security and mitigate vulnerabilities during the design phase. The proposed framework sets out a software security estimation process consisting of the following phases (Fig. 1 shows a diagram of the proposed process): 1) Identify the security factors; 2) Identify/Design the set of metrics; 3) Validate the set of metrics; 4) Quantify the security factors; 5) Estimate the security.

2.3 Common Criteria or ISO/IEC 15408 [11]

The Common Criteria (CC) which are currently standardized under the series of ISO/IEC 15408 standards came into being in 1990 as a result of the harmonization of criteria concerning the security of software products already used in various countries, with the intention that the result of the evaluation process could be accepted in numerous countries. The CC allow the results of independent evaluations of products to be compared. This is done by proposing a common set of functional requirements for IT (Information Technology) products. These products can be hardware, software or firmware. The evaluation process establishes a level of confidence in the degree to which the IT product satisfies the security functionality of these products and has surpassed the evaluation measures applied. The CC are useful as a guideline for the development, evaluation and acquisition of any IT products that include a security function.

In order to certify a product in accordance with the CC it is necessary to verify, via an approved independent laboratory, numerous security parameters upon which 22 countries worldwide have reached a consensus and have approved. The evaluation process includes the certification of a specific software product, in which the following aspects are verified:

- The product’s requirements are correctly defined.
- The requirements are correctly implemented.
- The product’s development process and documentation fulfil certain previously established requirements.

The CC thus establish a set of requirements with which to define the security functions of IT products and systems, along with the criteria used to evaluate their security. The evaluation process, which is carried out in accordance with the rules established by the CC, guarantees that the security functions of these products and systems meet the necessary requirements. Customers can therefore specify a product’s security functionality in terms of standard protection profiles, and can independently select the level of confidence in the evaluation of a set defined from EAL1 to EAL7.

The confidence levels in the evaluation defined in ISO/IEC 15408 vary from EAL1 (the lowest) to EAL7 (the highest), and are defined in an accumulative manner (verifications of level n+1 imply carrying out those of level n, 1…).

The EAL levels from 5 to 7 include semi-formal and formal models and demonstrations, and are therefore applied to products with very specific security objectives, such as those in the military sphere. These levels also require the generation of a large amount of documentation during the development process which must be handed to the evaluator to permit the information contained in them to be confirmed. Finally, in order for the Common Criteria to be applied, there exists a methodology containing the criteria needed to evaluate each of the levels of confidence standardised by the ISO/IEC 18045 (ISO 18045, 2008) standard, and denominated as CEM (Common Methodology for IT Security Evaluation).

2.4 ISO/IEC 27004 [12]

The family of ISO/IEC 27000 standards is composed of a set of documents, all of which are related to security management. 27000 specifically includes the definition of a common vocabulary concerning security management, 27001 provides a model with which to establish, implement, operate, control, review and maintain information security management systems, 27002 offers a code of good practices, 27003 offers implementation guidelines, 27004 is related to metrics for security management, 27005 deals with risk management, 27006 shows a body for the identification of security, and 27007 offers auditing guidelines. This family of standards (which is still incomplete) represents an effort to group together and unify those standards which are relative to security management, and is intended to be a reference model in the future.

The use of the ISO/IEC 27004 standard allows organisations to answer those questions related to how effective and efficient their ISMS (Information Security Management System) is and which levels of implementation and maturity have been reached. These measures will allow the organisations to compare the achievements made in information security over periods of time in business areas similar to their own, and as part of a continuous improvement.

This standard defines the scope, as a guide to the specification and use of measurement techniques, in order to provide precision in the observance of the ISMS in any type of organisation, with the aim of creating a basis with which to collect, analyse and communicate data related to this ISMS, which will be used to make decisions to improve it..

It is based on the PDCA (Plan – Do – Check – Act) model, which is a continuous cycle. This could be resumed in the idea that the measures are principally oriented towards "Do" (the Implementation and operation of ISMS), as an entrance for the "Check" (Monitor and review), which will thus allow improvement decisions concerning the ISMS to be adopted through "Act". The standard establishes that an organisation must describe how the ISMS and the measures inter-relate and interact, and must develop guidelines which ensure, clarify and document this relationship in as much detail as possible. It must also develop a programme regarding how the information security measure will be developed. The success of this programme will be based on the assistance or help of the measures provided in the decision making process. This measure programme must therefore be based on an information security measure "Model". The regulation also specifies how to develop the measures in order to be able to quantify the efficiency of an ISMS, its processes and its controls. The measures must be completely integrated into the ISMS.

2.5 An approach to measuring a system's attack surface [16]

In this approach the authors propose a metric with which to determine whether one software system is more secure than another similar one with regard to its attack surface. They use a measurement of a system's attack surface as an indicator of its security, signifying that the greater the attack surface is, the less secure the system will be. The system's attack surface is measured in terms of the three types of resources used in system attacks: methods, channels and data. The authors also demonstrate the use of their attack surface metric by measuring the attack surfaces of two IMAP servers and open source FTP demons. .

The attack surface is measured by following the three steps shown below:

1. Given a system *s*, and its environment, *Es*, the set, *M*, is defined from the entrance and exit points, a set, *C*, is defined from the channels, and a set, *I*, is defined from *s*'s untrustworthy data items.
2. The damage to the force-potential ratio, derm(m), is estimated for each method m$\in$M, the damage to the force-potential ratio, derc(c), is estimated for each channel c$\in$C, and the damage to the force-potential ratio, derd(d), is estimated for each data item d$\in$I.
3. *S*'s attack surface metric is the triple

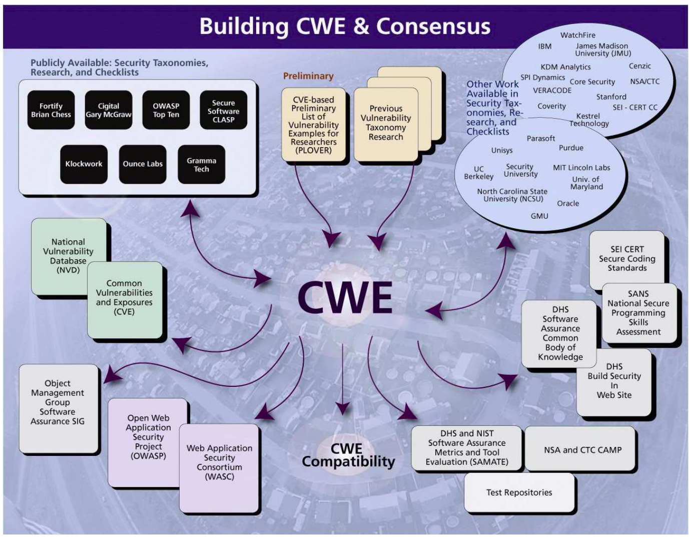
$$\langle \sum_{m \in M} der_m(m),\ \sum_{c \in C} der_c(c),\ \sum_{d \in I} der_d(d) \rangle$$

This metric can be used by software developers as a tool in the software development process, and by software consumers in the decision making process.

2.6 CWE (Common Weakness Enumeration) [18]

CWE provides a set of unified and measurable software weaknesses which facilitate an effective discussion, description, selection and use of software security services and tools, thus permitting these weaknesses to be discovered in the source code or in operational systems, along with facilitating a better understanding and management of those software weaknesses related to architecture and design. Fig. 2 shows a schema of the construction of CWE and how the consensuses are established

Fig. 2 CWE construction and consensus


Building CWE & Consensus
Publicly Available: Security Taxonomies, Research, and Checklists
Fortify Brian Chess
Cigital Gary McGraw
OWASP Top Ten
Secure Software CLASP
Klockwork
Ounce Labs
Gramma Tech
Preliminary
CVE-based Preliminary List of Vulnerability Examples for Researchers (PLOVER)
Previous Vulnerability Taxonomy Research
Other Work Available in Security Taxonomies, Research, and Checklists
WatchFire
IBM
James Madison University (JMU)
KDM Analytics
Cenzic
SPI Dynamics
Core Security
NSA/CTC
VERACODE
Stanford
Coverity
SEI - CERT CC
Kestrel Technology
Parasoft
Unisys
Purdue
UC Berkeley
Security University
MIT Lincoln Labs
Univ. of Maryland
North Carolina State University (NCSU)
Oracle
GMU
National Vulnerability Database (NVD)
Common Vulnerabilities and Exposures (CVE)
CWE
SEI CERT Secure Coding Standards
SANS National Secure Programming Skills Assessment
DHS Software Assurance Common Body of Knowledge
DHS Build Security In Web Site
Object Management Group Software Assurance SIG
Open Web Application Security Project (OWASP)
Web Application Security Consortium (WASC)
CWE Compatibility
DHS and NIST Software Assurance Metrics and Tool Evaluation (SAMATE)
NSA and CTC CAMP
Test Repositories

2.7 CVSS (Common Vulnerability Scoring System) [19]

CVSS is a public initiative conceived by the National Infrastructure Assurance Council (NIAC) in the USA, a group which puts into practice recommendations made by the same country's. Among those organizations that adopted CVSS at an early stage we can highlight Cisco, US National Institute of Standards and Technology (NIST), Qualys and Oracle. CVSS is currently in the custody of the Forum for International Response Teams (FIRST).

Among the benefits offered by the CVSS are:

- **Standardized punctuation of vulnerabilities:** The CVSS is neutral as regards applications, thus permitting different organisations to score their IT vulnerabilities via a single schema..

- **Contextualised score:** The score assigned by the organisation corresponds with the risks that the vulnerabilities represent for that organisation..
- **Open system:** The CVSS provides all the details concerning the parameters used to create each score, thus permitting organisations to understand both the reasoning behind a score and the significance of differences between scores.

The scores assigned by the CVSS are derived from the following three groups of metrics:

- **Base:** This group represents the properties of a vulnerability which do not alter over time, specifically: complexity of access, access vector, and degree to which the system's confidentiality, integrity and availability are compromised.
- **Temporal:** This group measures the properties of a vulnerability which alter over time, such as the existence of patches or code which could be exploited..
- **Environmental:** This group measures the properties of a vulnerability which are representative of the environment in which the IT is used, such as the prevalence of affected systems and potential losses.

The CVSS uses simple formulas along with the groups of metrics shown above to produce the final score associated with the vulnerability.

The **base metrics** are used to derive a score from 0.0 to 10.0 based on the responses to the following questions:

- Exploitability Metrics
  - Access vector: Local; Remote.
  - Attack complexity: High; Low.
  - Level of authentication needed: Not needed; Needed.
- Impact Metrics
  - Confidentiality impact: None; Partial; Complete
  - Integrity impact: None; Partial; Complete
  - Availability impact: None; Partial; Complete
  - Impact Bias: Confidentiality; Integrity; Availability

The Temporal Metrics modify the score base, reducing it by up to a third depending on the responses to the following questions:

- Exploitability:
  - Not tested; Attack prototype; Existence of exploitation; High
- Remediation level:
  - Official solution; Temporary solution; Contingency solution; Not available
- Report Confidence

Finally, the **environmental metrics** modify the score obtained and generate a final value depending on the responses to the following questions:

- Collateral damage potential:
  - None; Low; Medium; High
- Target distribution:
  - None; Low; Medium; High

The CVSS was designed in such a way that it would be understandable to the general public, and to permit any organisation to prioritize the order in which it wishes to tackle computing vulnerabilities that affect it, regardless of the technology used by that organisation in its computing systems. The principal advantage of the CVSS is that it resolves the problem of multiple vulnerability evaluation systems which are usually owned by the company and are incompatible with each other. Among its strengths we can highlight elegance, precision, flexibility and relative simplicity.

As with all vulnerability evaluation systems, the CVSS has its limitations. For example, it does not provide mechanisms with which to incorporate individual scores via various computing systems or organisational units. For this reason alone it is not appropriate for IT risk management since it does not consider mitigation strategies such as the installation of firewalls or access control procedures. Neither is it a score repository, as is Bugtraq, a vulnerabilities data base, as is the Open Source Vulnerability Database, or a vulnerabilities classification system as are Common Vulnerabilities and Exposures. However, the CVSS is relevant because it eliminates the duplicity of effort in the evaluation of IT vulnerabilities and it allows organisations to make decisions with more or less information.

2.8 CMSS (Common Misuse Scoring System) [23]

CMSS is an open scoring scheme which is standardised to measure the severity of software characteristic misuse vulnerabilities. Software characteristic misuse vulnerabilities are those vulnerabilities in which the characteristic provides a means to compromise the system's security. These vulnerabilities thus allow attackers to make malicious use of the supposedly beneficial functionality for which these characteristics were created.

CMSS is related to CVSS and CCSS (Common Configuration Scoring System), which are methods with which to score security breaches in software and configuration, respectively. The three standardised scoring systems permit the long term comparison of analyses carried out by different people and businesses. CMSS is derived from CVSS.

The scores assigned by the CMSS are derived from three groups of metrics: base, temporal and environmental. The base metrics are used to evaluate the intrinsic exploitability of the vulnerability and the impact on the confidentiality, integrity and availability. The temporal measures measure the aspects of variation in time of the severity of the vulnerabilities, such as the preponderance of exploits. The environmental metrics measure those aspects of vulnerability related to the specific vulnerability of the organisation's environment, such as the local implementation of countermeasures. The CMSS also includes a formula which combines these measures in order to provide a score for the severity of each vulnerability.

CMSS makes it easier for organisations to make decisions based on a quantitive standardised evaluation of software characteristic misuse vulnerabilities.

2.9 NIST 800-55 Security Metrics Guide for Information Technology Systems [25]

The NIST 800-55 regulation is a guide for security metrics published by the NIST (National Institute of Standards and Technology, in the USA). This guide assists in the development, selection and implementation of measures to be used at the level of information systems and programmes. These measures indicate the effectiveness of the security controls applied to the information systems which support security programmes. These measures enable decisions to be made, improve output and increase traceability through the collection, analysis and reports of related output data, thus providing a means by which to allow the organization to successfully balance the implementation, efficiency and effectiveness of the information system and programme security controls. The output measurement development programme described in this guide assists the organisation's information security experts to establish the relationships between the system's security activities, the programmes in its scope and the organisation's mission, thus helping to demonstrate the value of information security in organisations.

This document is centered on the development and collection of three types of metrics:

- Implantation metrics with which to measure the execution of the security policy.
- Effectiveness/efficiency metrics with which to measure the execution of the security policy.
- Impact metrics with which to measure the consequences of the security event for the business or its mission.

The types of measures that can really be obtained and which may also be useful in improving output depend on the maturity of the organisation's information security programme and the implementation of the security controls in its information systems. Although different types of metrics can be used simultaneously, the main focus of the information security metrics changes as the implementation of the security controls matures.

2.10 Security metrics for software systems [27]

The authors describe an approach with which to define software security metrics based on vulnerabilities included in software systems and their impact on software quality. This approach is based on CVSS and CVE (Common Vulnerabilities and Exposures), an industrial standard for vulnerabilities and revelation names.

The authors propose the following metrics:

$$SM(s) = \sum_{n=1}^{m} \left(P_n \times W_n\right), \qquad (1)$$

$$W_n = \frac{\sum_{i=1}^{K} V_i}{K} \qquad (2)$$

$$P_n = \frac{R_n}{\sum_{i=1}^{m} R_i} \qquad (3)$$

$$R_n = \frac{K}{M} \qquad (4)$$

$$\sum_{n=1}^{m} P_n = 1 \qquad (5)$$

Where *SM(s)* represents the security metric for the software *s*, and *Wi* (i = 1, 2, …, m) are the severity of those weaknesses that are representative of the software *s*. If we suppose that the weakness corresponds to *Wn* then it has *k* vulnerabilities and, according to the CVSS, its corresponding base scores are *V1, V2, …, Vk*. The severity of this weakness, *Wn,* is defined as their mean score, as is shown in formula (2). In formula (1), each *Pi* (i = 1, 2, …, m) represents the risk of each corresponding weakness. The percentage of occurrence of each representative weakness in the total number of occurrences is used to calculate *Pi* , as is shown in formula (3), where *Rn* is the frequency of occurrence of each representative weakness over a period of months, as is shown in formula (4), where *K* is the number of weaknesses and *M* is the number of months. Formula (5) is needed to normalize *Pn* in order to obtain the value of the metric *SM(s)* in a range of 1 to 10.

3. STUDY OF THE METRICS

Firstly, and as a previous step towards the construction of a software design security metrics model, we have analysed the different characteristics, sub-characteristics and sub-sub-characteristics related to security that are present in the various metrics models considered in the previous section. This has been done through the construction of Table 2, which shows the various security properties in the approaches analysed (these approaches have been numbered for greater legibility, and their correspondence is shown in Table 1). The crosses specify the relationship between each of these properties with and each approach. There is a blank space if the approach does not consider the proposal, a "P" if it considers it partially, and an "X" if it is clearly considered as part of the approach.

The reason for considering certain properties as characteristics, sub-characteristics or even sub-sub-characteristics is basically justified by two facts. The first of these involves the changes which has taken place in the last few years in how security is considered (this justifies the variation between ISO/IEC 9126 and ISO/IEC 25010). The second is the orientation of the proposals, since those that consider security as a sub-characteristic are those approaches that consider security in a general manner, whilst those that consider these properties as characteristics do so because they are approaches which are clearly oriented towards security

We have therefore constructed a canonic group of characteristics based on the security quality model proposed by [7], which serves as a basis through which to compare the aforementioned security characteristics from the perspective of the measurement of security in software design. That is to say, in Table 2 we compare how the previously described approaches measure this group of security characteristics from the perspective of software design.

An analysis of Table 2 shows that it is difficult to cover all the security properties from the design perspective, since it will be noted that the different approaches for design security metrics attempt to cover certain security characteristics at the design level, but do not cover all the security characteristics of the model if we use the comparison as a basis. Likewise, the majority of these approaches centre on general security metrics and are, in general, only really applicable in phases which are subsequent to the software design stage. Many of them must therefore be adapted if they are to be applicable from the point of view of software design, bearing in mind that the sooner and the better they can measure the security, the more economical whilst simultaneously robust the information system will be.

Table 1 Table showing correspondence of approaches

No. Approach	Name of Approach
1	Security metrics for object-oriented class designs [1]
2	Security estimation framework: design phase perspective [3]
3	Common Criteria [11]
4	ISO/IEC 27004
5	An approach to measuring a system’s attack surface [16]
6	CWE
7	CVSS
8	CMSS
9	NIST 800-55
10	Security metrics for software systems [27]

Table 2. Comparative table summarising approaches

Characteristic \ Approach	1	2	3	4	5	6	7	8	9	10
Authenticity	X		X	X	P	X	X	X	X	P
Confidentiality	X		X	X	P	X	X	X	X	P
Conformance		X	X	X	P	X	X	X	X	P
Detection of Attacks	P		P	P	X	X	X	X	P	P
Availability				P		X	X	X	P	P
Integrity	X		X	X	P	X	X	X	X	P
No Repudiation			X	P		P	P	P	P	P
Traceability	P	X	X	X		P	P	P	X	P
Conformance (*safety*)		X	X	X		X	X	X	X	P
Security and health of operator				P		P	P	P	P	P
Public health and security				P		P	P	P		
Commercial damage						P	P	P	P	P
Environmental damage						P	X	P		

We therefore consider that the development of a software design security metrics model is an area of knowledge which has yet to be developed, and that this should be a research objective if advances are be made in this subject. Moreover, the aforementioned security metrics model should be aligned with a security quality model which would serve as a reference both for security standards and to permit the complete measurement of software design security, i.e., by considering all the characteristics of the security quality model and the corresponding security standards. For example, the metrics should be aligned with the ISO/IEC 27004 standard, fundamentally with regard to the ISMS (Information Security Management Systems) security measurement.

4. CONCLUSIONS

Several approaches have already appeared whose intention it is to tackle security systematically together with the development of software products [2, 5, 8, 9, 14, 15, 20, 22]. There has also been an emergence of regulations and standards for security metrics such as those previously described, along with works based on the scoring of vulnerabilities and weaknesses, such as (CVSS [19], CMSS [23], CWE [18]), those based on the analysis of source code [10], those based on system architecture security measurement [16], those based on the measurement of security in object oriented class diagrams [1], or those based on risk ([24], or MAGERIT [17]). However, although many of these approaches are extremely interesting, they habitually deal with security in a partial manner, and do not clearly follow the aspects of security throughout the development process, thus leading to a situation in which the definition of security metrics at the design level has received little attention in recent years [1].

It is therefore necessary to define a set of metrics, both at the design level and later, which is related to the implementation level and which will allow us to evaluate the fulfilment level needed by security requirements which have been specified in the software analysis stages. These metrics must, moreover, be integrated into a security model (as a quality component) which has a clearly identified taxonomy of security requirements for which they can be identified, modelled and implemented, along with the remaining requirements, be they functional or non-functional.

In future works we shall use the approaches analysed here as basis to propose both a security model and a design security metrics model. These models will be a concept integrated approach whose intention will be to offer a common vision of the area, both with regard to characteristics and sub-characteristics and to their formal definition.

5. ACKNOWLEDGMENTS

This research is part of the following projects: MEDUSAS (IDI-20090557), financed by the Centre for Industrial Technological Development (CDTI), BUSINESS (PET2008-0136) awarded by the Spanish Ministry for Science and Technology and SEGMENT (HITO-09-138) and SISTEMAS (PII2I09-0150-3135) financed by the Council of Education and Science of the Castilla-La Mancha Regional Government.

6. REFERENCES

1. B. Alshammari, C. Fidge, and D. Corney, *Security Metrics for Object-Oriented Class Designs.* 2009 Ninth International Conference on Quality Software, 2009: p. 11-20.
2. D. Basin, J. Doser, and T. Lodderstedt, *Model Driven Security: from UML Models to Access Control Infrastructures.* ACM Transactions on Software Engineering and Methodology, 2006. **15**(1): p. 39-91.
3. S. Chandra, R.A. Khan, and A. Agrawal, *Security Estimation Framework: Design Phase Perspective.* 2009 Sixth International Conference on Information Technology - New Generations, 2009: p. 254-259.
4. K.-K.R. Choo, R.G. Smith, and R. McCusker, *Future directions in technology-enabled crime: 2007–09*, in *Research and Public Policy Series*, Australian_Government, Editor. 2007, Australian Institute of Criminology.
5. E. Fernandez-Medina and M. Piattini, *Designing Secure Databases.* Information and Software Technology, 2005. **47**(7): p. 463-477.
6. FIPS, *FIPS 140-2*, in *Security Requirements for Cryptographic Modules.* 2001, Federal Information Processing Standardization - National Institute of Standards and Technology

7. A.E. Fornaris, L.E. Sánchez, and E. Fernández-Medina, *Modelo de calidad para la seguridad.* RECSI (X Reunión Española sobre Criptología y Seguridad de la Información), 2010: p. (submitted).
8. C. Gutiérrez, E. Fernandez-Medina, and M. Piattini, *Towards a Process for Web Services Security.* Journal of Research and Practice in Information Technology, 2006. **38**(1): p. 57-67.
9. M. Hafner, R. Breu, B. Agreiter, and A. Nowak, *SECTET: An Extensible Framework for the realization of Secure inter-organizational Workflows.* Internet Research, 2006. **16**(5): p. 491-506.
10. I. Chowdhury, B. Chan, and M. Zulkernine, *Security metrics for source code structures*, in *Proceedings of the Fourth International Workshop on Software Engineering for Secure Systems*. 2008, ACM: Leipzig, Germany.
11. ISO/IEC, *ISO/IEC 15408:2005 Information technology - Security techniques - Evaluation criteria for IT security, (Common Criteria v3.0)*. 2005.
12. ISO/IEC, *ISO/IEC 27004:2009 - Information technology -- Security techniques -- Information security management -- Measurement.* 2009.
13. ITU, *ICT Security Standards Roadmap* 2009, International Telecommunication Union.
14. J. Jürjens, *UMLsec: Extending UML for secure systems development*, in *UML 2002 - The Unified Modeling Language, Model engineering, concepts and tools*, J. Jézéquel, H. Hussmann, and S. Cook, Editors. 2002, Springer. LNCS 2460.: Dresden, Germany. p. 412-425.
15. J. Jürjens, *Secure Systems Development with UML*. 2004: Springer-Verlag.
16. P.K. Manadhata, K.M.C. Tan, R.A. Maxion, and J.M. Wing, *An Approach to Measuring A System's Attack Surface*. 2007, Carnegie Mellon University: Pittsburgh,.
17. MAP, *Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT - v 2)*. 2005, (Ministry for Public Administration of Spain).
18. R.A. Martin, *Common Weakness Enumeration (CWE v1.8)*. 2010, National Cyber Security Division of the U.S. Department of Homeland Security.
19. P. Mell, K. Scarfone, and S. Romanosky, *A Complete Guide to the Common Vulnerability Scoring System (CVSS 2.0)*. 2007, NIST and Carnegie Mellon University.
20. D. Mellado, E. Fernandez-Medina, and M. Piattini, *A Common Criteria Based Security Requirements Engineering Process for Development of Securie Information Systems.* Computer Standards & Interfaces, 2006. **29**(2): p. 244-253.
21. A.L. Opdahl and G. Sindre, *Experimental comparison of attack trees and misuse cases for security threat identification.* Information and Software Technology. In Press, Corrected Proof, 2008.
22. A. Rodríguez, E. Fernández-Medina, and M. Piattini. *M-BPSec: A Method for Security Requirement Elicitation from a UML 2.0 Business Process Specification*. in *3rd International Workshop on Foundations and Practices of UML*. 2007. Auckland, New Zealand.
23. E.V. Ruitenbeek and K. Scarfone, *The Common Misuse Scoring System (CMSS): Metrics for Software Feature Misuse Vulnerabilities*, in *NIST Interagency Report 7517*. 2009, National Institute of Standards and Technology.
24. O.S. Saydjari, *Is Risk a good security metric?* Quality of Protection Workshop – Security Measurements and Metrics (QoP’06), 2006: p. 59-60.
25. M. Swanson, N. Bartol, J. Sabato, J. Hash, and L. Graffo, *Security Metrics Guide for Information Technology Systems*, in *NIST Special Publication 800-55 Revision 1*. 2008, National Institute of Standards and Technologies.
26. A.J.A. Wang, *Information Security Models and Metrics.* 43nd ACM Southeast Conference, 2005: p. 2-178 to 2-184.
27. J.A. Wang, H. Wang, M. Guo, and M. Xia, *Security Metrics for Software Systems*, in *Proceedings of the 47th Annual Southeast Regional Conference (ACMSE '09)*. 2009.