WOSIS 2011

David G. Rosado, Luis Enriques Sánchez and
Jan Jürjens (Eds.)

Security in Information Systems

Proceedings of WOSIS 2011
8th International Workshop on Security in Information Systems
In conjunction with ICEIS 2011
Beijing - China, June 2011

David G. Rosado
Luis Enrique Sánchez and
Jan Jürjens (Eds.)

Security in Information Systems

Proceedings of the
8th International Workshop on
Security in Information Systems
WOSIS 2011

In conjunction with ICEIS 2011
Beijing, China, June 2011

SciTePress
Portugal

Volume Editors

David G. Rosado
University of Castilla-la Mancha
Spain

Luis Enrique Sánchez
University of Castilla-La Mancha
Spain

and

Jan Jürjens
Technical University of Dortmund
Germany

8th International Workshop on
Security in Information Systems
Beijing, China, June 2011

Printed in China

ISBN: 978-989-8425-61-4
Depósito Legal: 327884/11

Foreword

The Eighth International Workshop on Security in Information Systems – WOSIS 2011 was organized in conjunction with ICEIS 2011 in Beijing, China. As in previous years, this workshop is primarily focused on high quality and innovative research papers from different fields related to the most recent developments in Security in Information Systems. Traditionally the best papers are published in a reputable journal dealing with WOSIS topics. This year, authors will have the opportunity to have their work selected for publication in an extended version in the well recognized ISI ranked Publication Journal of Universal Computer Science.

In this new edition, Dr. Shareeful Islam will honour us with his great experience offering the Keynote Speech of WOSIS 2011. Shareeful will speak us about a framework which supports alignment of secure software engineering with legal regulations. We want to acknowledge his contribution which we hope you find motivating.

Papers presenting the most recent theoretical, and practical works in security for Information Systems were received, a total of 27 submissions. This year the number of submitted papers has considerably increased, maybe due to the incorporation of new topics and having the backing of a prestigious journal. All the submissions were reviewed by at least two program committee members. Finally, 10 papers have been accepted as full papers and 10 short papers will also have the chance to be presented during the sessions due to the excellent quality of the research.

We would like to thank all the authors who took the time to submit papers to WOSIS, even though they were not finally accepted. Because of the high quality of the work submitted the review process was very difficult, and some good papers had to be rejected because of the high volume of work received. We would also to express our gratitude for the excellent work done by the Program Committee and the members of the Organisation Committee.

The publication of the best papers in the prestigious journal of Universal Computer Science, along with the presence of a renowned Program Committee and Keynote Speaker, will contribute to the success of this 8th edition of WOSIS.

June 2011,

David G. Rosado
University of Castilla-la Mancha, Spain

Luis Enrique Sánchez
University of Castilla-La Mancha, Spain

Jan Jürjens
Technical University of Dortmund, Germany

Workshop Co-chairs

David G. Rosado
University of Castilla-la Mancha
Spain

Luis Enrique Sánchez
Sicaman Nuevas Tecnologias S.L.
Spain

and

Jan Jürjens
Technical University of Dortmund
Germany

Invited Speaker

Shareeful Islam
Lecturer School of Computing, IT and Engineering University of East London
United Kingdom

Dr. Shareeful Islam was awarded his PhD in Software Risk Management Model using goal-driven approach from chair of Software & Systems Engineering (I4), Technische Universität München, Germany. He has received M.Sc. degree in Information Communication System Security(ICSS) from the Royal Institute of Technology, Sweden. He also received M.Sc. degree in Computer Science (CS)and B.Sc. (Hon's) in applied physics and electronics(APE) from the University of Dhaka, Bangladesh. He completed the ISO 9001:2001 lead auditor certification and is a certified quality management system auditor. He has more than 10 publication in well recognized journals and conferences like Requirements Engineering Journal (REJ) and Journal of Software and Systems Modeling (SoSyM), REFSQ ESEC/ FSE . His main research interests are in the field of software risk management, software security and privacy. Special interests are risk management model, security and privacy, requirements engineering and modeling.

Program Committee

Ernesto Damiani, Università degli Studi di Milano, Italy
Jaime Delgado, Universitat Politècnica de Catalunya, Spain
Csilla Farkas, University of South Carolina, U.S.A.
Stefan Fenz, Vienna University of Technology, Austria
Eduardo B. Fernandez, Florida Atlantic University, U.S.A.
Maria Carmen Fernández, University of Málaga, Spain
Eduardo Fernández-medina, University of Castilla-La Mancha, Spain
Steven Furnell, University of Plymouth, U.K.
Carlos Gutierrez, Correos Telecom, Spain
Michael Hafner, University of Innsbruck, Austria
Renato Iannella, Semantic Identity, Australia
Stamatis Karnouskos, Sap, Germany
Jaejoon Lee, Lancaster University, U.K.
Fabio Massacci, Dep. of Information and Communication Technology, Italy
Raimundas Matulevicius, University of Tartu, Estonia
Haralambos Mouratidis, University of East London, U.K.
Brajendra Panda, University of Arkansas, U.S.A.
Günther Pernul, University of Regensburg, Germany
Mario Piattini, Escuela Superior de Informatica, Spain
Joachim Posegga, Inst. of IT Security and Security Law, Germany
Sasa Radomirovic, University of Luxembourg, Luxembourg
Indrakshi Ray, Colorado State University, U.S.A.
Alfonso Rodriguez, University of Bio-Bio, Chile
Thomas Santen, European Microsoft Innovation Center, Germany
Ketil Stoelen, Sintef, Norway
Ambrosio Toval, University of Murcia, Spain
Sabrina de Capitani di Vimercati, Università degli Studi di Milano, Italy
Toshihiro Yamauchi, Okayama University, Japan

Table of Contents

Full Papers

Short Papers

Towards a Pattern-Based Security Methodology to Build Secure Information Systems

Roberto Ortiz[1], Santiago Moral-Rubio[1], Javier Garzás[2, 3] and Eduardo Fernández-Medina[4]

[1] Dep. Information Security. BBVA Group, Madrid, Spain
r.ortizpl@gmail.com; santiago.moral@bbva.com
[2] Kybele Group. Dep. of Computer Languages and Systems II
University Rey Juan Carlos, Madrid, Spain
javier.garzas@urjc.es
[3] Kybele Consulting, Madrid, Spain
javier.garzas@kybeleconsulting.com
[4] GSyA Research Group. Dep. of Information Technologies and Systems
University of Castilla-La Mancha, Ciudad Real, Spain
Eduardo.FdezMedina@uclm.es

Abstract. Methodologies for the construction of secure systems provide a controlled, planned development process, with verifications in all stages, thus avoiding unexpected errors and leading to an improvement in the quality and security of the system produced. These methodologies can be enriched from the use of security patterns, since these tools are widely accepted by both the scientific community and industry for the construction of secure information systems owing to the fact that they accumulate security experts' knowledge in a documented and structured manner, thus providing a systematic means to solve recurrent problems. In this paper we present a first approximation of a pattern-based security methodology to support both the construction of secure information systems and maintenance of the level of security attained. This proposal is based on real case studies, and is now in the first stages of application in real settings. Interesting results are already appearing that will allow us to refine and validate the proposal.

1 Introduction

The importance of the design of secure systems has increased because the majority of attacks on Information Systems (IS) are based on vulnerabilities caused by deficiencies in their design and in the development functionalities with which these systems are equipped [4].

The use of methodologies for the construction of this type of systems plays an important role in obtaining a secure IS, since they provide a systematic, planned, controlled, verifiable and thorough development process, thus avoiding the existence of risks which go unnoticed, are omitted, or are badly communicated to the rest of the system [16]. The use of methodologies in this field will therefore have a bearing on an improvement in the security of the system produced. This benefit is obtained owing to the fact that, among other things, the process used to add security to an IS is

decomposed to the level of elementary activities, in which each activity is identified by a procedure that defines the means to carry it out, the most appropriate actors for its implementation, and the tools and techniques needed [17].

The objective of security methodologies is to provide solutions to problems related to security vulnerabilities and thus minimize the impact of attacks on IS. Since the majority of problems occur in the similar way in different contexts, generic solutions to these problems can be expressed as patterns [3].

When constructing secure IS, a methodology might therefore be more complete if it takes advantage of security patterns [1], since these are useful tools with which to systemize the process in order to solve recurrent security problems owing to their provision of guidelines for the construction and evaluation of secure systems [15].

This proposal presents a first approximation of a pattern-supported methodology to build secure IS. We are conscious of the fact that more factors than just security patterns are involved in a methodology, such as heuristics, good practices, rules, etc., but in this first proposal we shall focus exclusively on the use of security patterns.

Our principal objective is to offer security engineers another systematic process, which is additional to the existing traditional Software (SW) development methodologies that it allows: building secure IS or maintaining the level of security attained in an organisation's systems.

The methodology that we propose is divided into stages. Each stage is, in turn, divided into activities. We shall also show the input artifacts, the output artifacts, and the technique, practice and reference guides used, along with the ideal principal roles to carry out each stage. Another of the main contributions of this methodology is that it revolves around a central axis, which is the criticality of the assets to be protected. The fundamental contribution of security patterns in this methodology is that they provide structured, validated and reusable security knowledge, both for experts and non-experts in security alike. Finally, we should stress that this methodology is based on real case studies, and is now in the first stages of use in a financial entity in which interesting results are already being obtained which will allow us to refine, test and validate it.

The remainder of this paper is organized as follows: in Section 2, we show the background of current security patterns, works and methodologies based on patterns. Section 3 describes the methodology itself, and the paper concludes with some conclusions in Section 4.

2 Security Patterns Background and Related Works

In the last year, a multitude of authors have researched about security patterns. Some of the most representative examples of the state of the art in this field are the works exposed in [21], [19], or [8]. Moreover, the use of security patterns as a guideline with which to design a secure IS is a fairly extended practise in industry, e.g., Microsoft and IBM use them [6, 12].

It is currently possible to find diverse proposals in which security is integrated into the construction of systems through the use of patterns. UMLsec [7], for example, extends UML to model security properties in informatics systems. This proposal has recently been extended in order to use patterns to support the modelling and

verification of formal aspects of security. In [5] the authors present a security engineering process based on Security Problem Frames and Concretized Security Problem Frames. These two types of frames constitute patterns with which to analyse security problems and associated solution approaches in SW developments. The above proposals are focused on the application of patterns in security systems, but centre solely on specific aspects such as the modelling of activities or security requirements, and not on the complete lifecycle of IS. Other proposals use patterns to deal with all the stages in the construction of secure IS. In [1, 2] the authors apply security patterns through the use of a secure system development method based on hierarchical architectures whose levels define the scope of each security mechanism. The main advantage of these works, which are an evolution of the same approach, is: the guidelines offered in each stage to assist the user in where to apply and how to select the security pattern which is most appropriate to satisfy the functional requirements or, to mitigate vulnerabilities in each stage. According to the authors of the aforementioned papers, one of their future works will be to implement this proposal in real environments. This aspect would greatly enrich the methodology, since it would permit the detection of underlying activities in the stages proposed and of the roles that intervene in each of them. It would also assist the authors to explore in greater depth a specification of security techniques which are more appropriate to carry out each of the activities in the stages.

In [18] the authors propose a method with which to integrate security patterns into a software engineering process. This proposal assists experts to close the gap between the abstract solution described in the pattern and the implementation proposed in the application. The cataloguing of different roles and the use of tools that support the systematic process is a valuable approach for real and complex organisations, but the complexity and dynamism of this type of entities makes it necessary to study the definition of additional specific security tasks in greater depth, in parallel to the SW development, in order to obtain secure systems.

3 Proposal for Security Methodology

The objective of our proposal is to offer security engineers a systematic process with which to construct secure IS and to maintain the level of security attained. This systematic process is enriched with the Know How of a team of security experts with wide experience in the application of security solutions in real complex systems.

The aforementioned systematic process is based on methodologies such as the Unified Process [11], in which a development and implementation process is carried out in an iterative and incremental manner. The advantages of this type of processes is that successive refinements can be carried out to identify critical risks and errors during the early stages by using testing mechanisms (which in our case are those of security) on the system in each of the stages to obtain a final effective solution.

The methodology that we propose is in parallel and additional to the traditional SW development process, and is divided into similar stages. At each stage, the mandatory input and output artifacts are specified. Fig. 1 shows the definition of this systematic process with the aforementioned characteristics through the use of SPEM (Software & Systems Process Engineering Metamodel) version 2.0 [14]. Each of

these stages is also composed of activities, and technique, practice and reference guides used in these activities, along with the ideal principal roles to carry out each stage. The main characteristic of the methodology is that its implementation revolves around the criticality of the assets to be protected, and that each of the stages is supported through the use of patterns which provide structured, documented, validated and reusable solutions to common security problems.

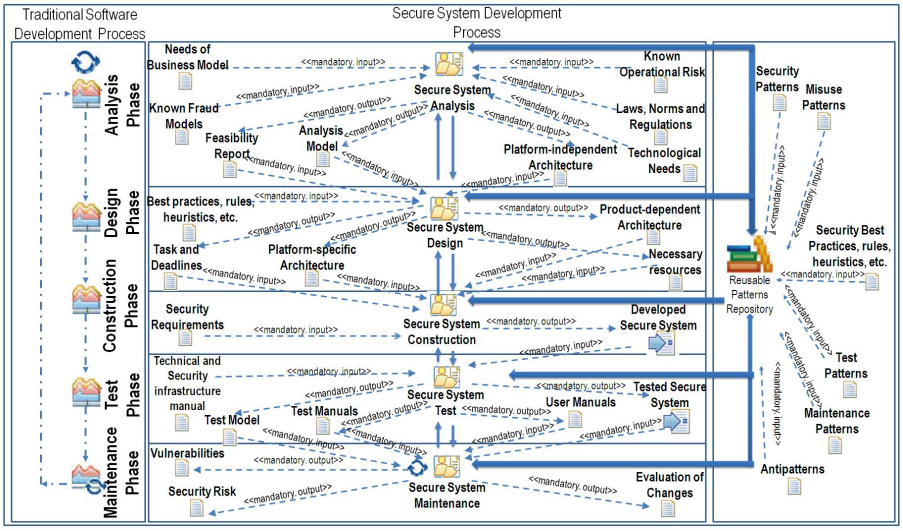

Fig. 1. Development Methodology for Secure Systems with SPEM 2.0.

The stages of the methodology are shown as follows:

Analysis Stage. The principal objective of this stage is to analyse the viability of the business model or project proposed. This is done by carrying out an iterative and incremental analysis to detect those risks that might affect the organization if the project is implemented, along with an in-depth analysis of the impact that this will have on the organisation's IS. Misuse cases, security use cases, attack trees, etc. which may affect the final solution are also detected.

Activities: This stage is composed of the following activities:

- A1: Initial Analysis: the project is analysed to identify the needs of the business model proposed. The objective, reach, the scope that will be affected by the proposal, current situation, future situation and expected benefits are identified.
- A2: Identification of needs. Identification of the actors involved, the systems that will be used, the processes derived, and, the needs to be covered.
- A3: Detection of assets to be protected. Here, we analyse those of the organisation's assets that will be involved in the business model proposed. A classification of these assets is carried out on the basis of their criticality in order to be able to establish the ideal means of protection to mitigate possible attacks.
- A4: Risk Analysis. This task is carried out to determine which risks might affect the project on the basis of the assets involved in relation to the following plans:

- Preventative (Fraud). An analysis of whether the implantation of the business model will have any consequences as regards known fraud models in the sector in which the organization operates.
- Legislative/Regulatory/Normative. This involves an evaluation of possible non-compliance with the existing legislation of the country where the organisation's IS are localized, of the regulations associated with the sector to which the organization belongs, or of the existing rules in the organization. For example, in a corporation dedicated to banking activities in Spain, the following would be evaluated:
  a. Legislative plan: Communications Secret, Banking Secret, LOPD, etc.
  b. Regulatory Plan: MIFIT, SOX, BASILEA, PCI, MAREA, etc.
  c. Normative Plan: the organisation's security policy.
- Operational. An analysis is made of whether the business model proposed includes risk of loss resulting from a lack of suitability or a fault in the processes, the personnel or the internal systems, or as a result of external events .

- A5: Risk evaluation. An evaluation of whether the risks detected are compatible with carrying out the project is performed. The risk detected is related to the potential impact that it will have on the organization, such as loss or degradation of image, economic losses, consequences in the business model, etc.
- A6: Identification of Security Patterns. Both the criticality of the assets to be protected and the detected risks are contrasted with the security patterns repository to discover whether or not known solutions to this type of problems exist. The patterns repository, in which the risks, impact and solutions will be related, will contain: patterns in the style of Misuse Patterns [3] which relate possible attacks or misuses with security means that will mitigate them; Antipatterns [9, 10] and patterns such as those shown in the template [13], which contain three solution levels to a specific security problem. There are three possible situations when consulting the repository: if it is a known need, then a known pattern will be applied; if it is a new need, but a pattern already exists which has solved similar problems, then this pattern only needs to be adapted; or if it is a new need then in-depth work will be necessary to provide a solution that will be converted into a new pattern by means of successive refinements. Whatever the solution is, the patterns repository will provide feedback with a new practical case. After consulting the repository we obtain an abstract security solution corresponding to the independent-platform level shown in [13].
- A7: Identification of technological requirements. The project's technological needs are detected on the basis of the abstract solution provided by the previous pattern. In this case the availability of data such as the number of users who access the system, these users' roles, where they access it from (inside or outside the organisation's security perimeter), treatment of assets accessed, time limit established for realization of the project, available budget, etc. is necessary.
- A8: Analysis of technological requirements. The requirements obtained are used to analyse the infrastructure and the available resources and, depending on the solution provided by the pattern, a study of which additional elements will be necessary is carried out. In this activity, we identify needs as being the dimensioning of systems (CPU, Memory, Storage, etc.), product licenses, the incorporation of new systems or personnel, software development, etc.

- A9: Evaluation of Technological Requirements. An estimation is made of the impact that the solution will have on the organisation's IS and its compatibility with them. The costs associated with the solution are also estimated.
- A10: Security viability Report. A report is produced which reflects the possibilities of undertaking the project from the point of view of security, and in which the following are stated: risks detected, the impact that these risks will have on the organization, the technological needs and their associated costs, platform-independent solution provided by the pattern, and the technical possibilities which can be used to tackle the project.
- A11: Review. The results from the report are reviewed in case new risks or new technological needs have appeared.
- A12: Acceptance of viability report. The results obtained are evaluated and a decision is made as to whether to scrap the project, or to undertake it assuming the risks detected and the costs associated with implementing the solution.

Input artifacts: Need of business model, Known Fraud Models, Known Operational Risk, Laws, Norms, Regulations, Technological needs, Security Patterns and Reusable Patterns Repository.

Output artifacts: Feasibility Report, Analysis Model, Platform-Independent Architecture.

Techniques, Practices and Reference guides. UML, UMLsec, risk analysis, Misuse Patterns, Security Patterns, misuse cases, security use cases, threat analysis.

Main Roles: Project Manager, Risk Analyst, Security Analyst, Security Requirements Engineer, Security Architect, Fraud Analyst, Legal Consultant.

Design Stage. The complete design of the security system revolves around the assets to be protected. The objectives of the means of security designed will be, on the one hand, to mitigate possible attacks that the systems may undergo, and on the other to reduce benefits that may be gained from these attacks.

Activities The activities in this stage are:

- A1: Definition and Design of Technological Architecture. The infrastructure of which the solution to the proposed business model will be composed will be extracted from the patterns repository and will correspond with the Platform-specific and Product-dependent levels shown in [13], which correspond with the security pattern selected in the Analysis Stage. This will depend on the technology which is available and on the security needs shown in the Analysis Stage. The security architecture defined will be used as a basis to extract the tasks to be carried out in order to implement this architecture, and each task is assigned to the most appropriate security expert.
- A2: Definition of Deadlines. The tasks that will be carried out by the different technical security groups will be planned to estimate the time needed to carry out the work designed.
- A3: Specification of Final Architecture (Security Report). A document is created which shows the architecture that will eventually be introduced and which certifies the exhaustive analysis of the proposed project carried out by the information security department. If new risks or threats to security are detected, iterations are carried out by returning to the previous stage until a secure design that covers these exceptions has been obtained.

- A4: Approval of Security Report. The document is analysed by the various security groups involved in order to ratify the work that they will carry out in the specified time limit and any new risks that may have arisen. This analysis will determine whether or not the project will go ahead.

Input artifacts: Output artifacts of Analysis Stage, Best Practices, rules, heuristics, etc., Security Patterns and Reusable Patterns Repository.

Output artifacts: Task and Deadlines, Necessary resources, Platform-specific architectures, and Product-dependent architecture provided by security pattern.

Techniques, Practices and Reference guides: Good design practices, good security practices, Security Patterns, Antipatterns, documentation.

Main Roles: Project Manager, Security Analyst, Security Experts, Security Developer, Security Architect.

Construction Stage. The objective of this stage is to construct the system proposed in the previous stage in a development setting. It is important to emphasize the need for segmentation of settings, and it is therefore obligatory for the entire infrastructure to be promoted through three clearly defined settings: development setting; unified test setting; and production setting (ultimate setting).

Activities: This stage is composed of the following activities:

- A1: Preparation of setting. The availability of all necessary resources, tools, infrastructure modules and personnel for the construction of the proposed system in the previous stage is checked.
- A2: Implementation of Security Architecture. The identification of the elements in the security infrastructure of which the system is composed, and the development of the architecture proposed in the Product-dependent level of the security pattern selected with the available mechanisms, tools and modules are carried out.
- A3: Implementation of security developments. The security programmes or cryptographic developments proposed by the pattern are developed. If these do not exist, then they are acquired from the market if the appropriate organization's tools or knowledge is not available.
- A4: Implementation and integration of final security architecture. The system is configured by integrating the security elements implemented and the security modules designed. The necessary communications are also implemented so that both the elements and the security developments are connected and thus carry out the security tasks defined by the security pattern.
- A5: Definition of Maintenance Patterns. The patterns repository is enriched to define the maintenance tasks, in pattern form, associated with a particular security pattern, if they have not been previously defined. The following will be defined: how to carry out the exploitation of the elements in the infrastructure used; which procedures are appropriate for the maintenance of the system developed; guidelines concerning how to act in the case of system incidents or failures in the programming code; the ideal period in which to perform reviews of the architecture developed; and, guidelines concerning the monitoring of the systems involved.

Input artifacts: Output artifact of the Design Stage, security requirements, Security Patterns and Reusable Patterns Repository.

Output artifacts: Developed Secure System, Maintenance Patterns, and Security Patterns.

Techniques, Practices and Reference guides: Security Patterns, Software Patterns, Antipatterns and Best Practices [20].

Main Roles: Security Engineer, Security Architects, Security Expert, Security Developer, Integrator Engineer, and D&D Team.

Test Stage. After integrating the system's hardware (HW) and SW components, it is necessary to ensure that they function correctly and that they fulfil that which is indicated in the previous stage, before being handed over to the final user.

Activities: The activities in this stage are:

- A1: Design of operation tests. The guidelines concerning how to act are defined to verify that the functioning of the system which has been developed is correct (communication, performance, accessibility tests, etc.).
- A2: Execution of operation tests. The correct functioning of the system is verified, in addition to verifying that the pieces of HW and SW of which it is composed are well developed and configured, and that connectivity exists between them.
- A3: Design the Security Test Patterns. The reusable patterns repository is by relating the Security Patterns to their associated Security Test Patterns, if they do not yet exist. The following will be defined in these patterns: Test to be carried out; Planning of Test; Personnel who will carry out the Test; And tools, resources and mechanisms necessary to carry out the security Test in the infrastructure proposed by the pattern. The Test will be carried out on the basis of the vulnerabilities of the technology, the operative systems, the SW, the HW, etc. The tests will consist of ethical hacking, intrusion tests, error and code quality tests, fault tolerance, backup systems, verification of militarization of machines and operative systems, and verification of activity register logs.
- A4: Execution of security tests. The security tests defined in the previous stage are carried out to certify that the system is secure. These tests are carried out in the setting designed for this purpose – the unified test setting.
- A5: Evaluation of tests. The creation of a report containing the results of the tests that certify that the system which has been developed is secure. In the case of discovering any faults in the configuration, or in the code or vulnerability in the system, the infrastructure is reviewed in an iterative manner by following the activities from the previous stage.
- A6: Approval of the system developed. Once the system has been developed and both its functioning and security have been verified, the system is certified as being ideal to satisfy the business model proposed in the project. Later, the system will be promoted to the production setting and will be available to the final user.
- A7: Monitoring the system. The system is now monitored on the basis of the guidelines defined by the Maintenance Pattern in order to discover any future anomalies, faults, output problems, vulnerabilities or deficiencies occasioned by the passage of time and the evolution of both the HW and SW systems and the tactics and tools used by attackers.

Input artifacts: Output artifact of the Construction stage, Technical and Security infrastructure manual, Security Patterns and Reusable Patterns Repository.

Output artifacts: Test Model, Test Patterns, User Manuals, and Test Manuals and Tested Secure system.

Techniques, Practices and Reference guides: Monitoring, Ethical Hacking, Hardening, Test reviews, and Misuse Patterns, Test Patterns.

Main Roles: Security Analyst, Test Engineer, Security Architects, Fraud Prevention Team, Security Management and Operation Team, Security D&D Team.

Maintenance Stage. Once the system is functioning, tests should be carried out periodically to guarantee that the level of security attained has not diminished as a result of the following: New vulnerabilities; new regulatory requirements or laws; the evolution of the competition; and deficient performance in the execution of the processes in a system, which may be owing to the maintenance processes, the production improvement processes, the automation of processes to reduce costs, etc.

Activities: The activities in this stage are detailed as follows:

- A1: Design of security tests. Verification of the availability of mechanisms, tools and resources necessary to carry out the security tests defined in the Test Stage as security Test patterns.
- A2: Planning of tests. The execution of the tests is planned in order to establish a period of time in which these security tests can be developed. During this period of time it is necessary to ensure that the integrated test setting is available and operative so that no impact will be made on the system's functioning.
- A3: Execution of tests. The tests defined by the Security Test Pattern are carried out.
- A4: Evaluation of results. The results obtained are analysed to detect possible security faults. If any vulnerability is detected in the technology, code, operative system etc., the changes which will be necessary to solve this problem are analysed, along with the economic and time impact that will entail the solution.
- A5: Evaluation of changes. The proposal for changes is analysed and its execution is planned. The Patterns Repository, the Security Pattern which provided the solution containing the necessary changes, and the Maintenance and Test Patterns associated with it are also updated, and all the systems that have been implemented by following the solution proposed by this Security Pattern are modified.
- A6: Execution of changes. The necessary changes are made in order to resolve the vulnerabilities detected.
- A7: Monitoring the system. The monitoring of the system is activated in the same manner as in Activity 7 of the Test Stage.

Input artifacts: Output artifacts of the Test Stage and Reusable Patterns Repository.

Output artifacts: Vulnerabilities and new Security Risk, and Evaluation Changes.

Techniques, Practices and Reference guides: Monitoring, Ethical Hacking, Hardening, Test Patterns, Cost/ Time/ Personnel/ Resource analysis.

Main Roles: Project Manager, Security Analyst, Security Architects, Fraud Prevention Team, Security Management and Operation Team, Security D&D Team.

4 Conclusions

The objective of this work is to propose a systematic process which is additional to those that appear in traditional SW development methodologies, in order to assist security engineers to build secure IS or maintain the level of security attained in their organisations' IS.

To achieve this, we propose a security methodology to build secure IS in which the activities corresponding with the stages are supported by the use of security patterns, which provide the knowledge accumulated from security material in a structured, documented and reusable manner. Other advantages of this methodology in comparison to certain others that already exist in literature are: the principal axis is the criticality of the assets to be protected, since, depending on the level, it is more restrictive when applying security measures; the stages are divided into clearly differentiated activities; input artefacts, output artefacts, and Technique, Practice and Reference guides are introduced in each stage to allow the activities to be carried out; and the most suitable roles to carry out the tasks are specified in each stage. Finally, we should stress that this methodology is based on practice cases and is currently in the first stages of application in a large organization in the banking sector. This is producing interesting results which are allowing us to refine and validate the methodology.

Another line on which we are working is the formalisation of this methodology in the SPEM 2.0 meta-modelling language.

Acknowledgements

This research has been carried out in the framework of the following projects: MODEL-CAOS (TIN2008-03582/TIN) financed by the Spanish Ministry of Education and Science, SISTEMAS (PII2I09-0150-3135) and SERENIDAD (PEII11-0327-7035) financed by the "Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" and the FEDER, and BUSINESS project (PET2008-0136) financed by the "Ministerio de Ciencia e Innovación", Spain.

References

1. Fernandez, E. B. (2009). Security Patterns and A Methodology to Apply them. *Security and Dependability for Ambient Intelligence* (pp. 37-46).
2. Fernandez, E. B., Larrondo-Petrie, M. M., Sorgente, T. and VanHilst, M. (2006). Chapter 5, A methodology to develop secure systems using patterns. *Integrating security and software engineering: Advances and future vision* (pp. 107-126): IDEA Press.
3. Fernandez, E. B., Yoshioka, N. and Washizaki, H. (2009). *Modeling Misuse Patterns*. Paper presented at the ARES '09. International Conference on Availability, Reliability and Security.
4. Halkidis, S. T., Tsantalis, N., Chatzigeorgiou, A. and Stephanides, G. (2008). Architectural Risk Analysis of Software Systems Based on Security Patterns. *IEEE Transactions on Dependable and Secure Computing, 5*(3), 129-142.
5. Hatebur, D., Heisel, M. and Schmidt, H. (2007). *A Security Engineering Process based on Patterns.* Paper presented at the DEXA '07. 18th International Conference on Database and Expert Systems Applications.
6. IBM. (2011). Introduction to Business Security Patterns, An IBM White Paper.
7. Jürjens, J. (2004). *Secure Systems Development with UML*: Springer-Verlag.
8. Kienzle, D. M., Elder, M. C., Tyree, D. and Edwards-Hewitt, J. (2006). Security patterns repository, version 1.0.

9. Kis, M. (2002). *Information Security Antipatterns in Software Requirements Engineering.* Paper presented at the Pattern Languages of Programs Conference.
10. Král, J. and Zemlicka, M. (2009). *Popular SOA Antipatterns*. Paper presented at the Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, Athens, Greece.
11. Kruchten, P. (2000). *The Rational Unified Process: An Introduction*. Boston: Addison-Wesley.
12. Microsoft. (2011). Patterns & Practices: Web Service Security Patterns
13. Moral-García, S., Ortiz, R., Moral-Rubio, S., Vela, B., Garzás, J. and Fernández-Medina, E. (2010). *A New Pattern Template to Support the Design of Security Architectures*. Paper presented at the The Second International Conferences of Pervasive Patterns and Applications, Lisbon, Portugal.
14. OMG. (2008). Software & Systems Process Engineering Meta-Model Specification (SPEM) 2.0.
15. Ortiz, R., Moral-García, S., Moral-Rubio, S., Vela, B., Garzás, J. and Fernández-Medina, E. (2010). *Applicability of Security Patterns*. Paper presented at the The 5th International Symposium on Information Security (IS'10 - OTM'10), Crete, Greece.
16. Pressman, R. (2004). *Software Engineering: A Practitioner's Approach*: McGraw-Hill Science/Engineering/Math.
17. Roberts, T. (1999). Why can't we implement this SDM? *IEEE Software 16*(6), 70 - 71, 75. doi: 10.1109/52.805477
18. Sanchez-Cid, F. and Maña, A. (2008). *Serenity Pattern-Based Software Development Life-Cycle*. Paper presented at the 19th International Workshop on Database and Expert Systems Application, 2008. DEXA '08., Turin.
19. Schumacher, M., Fernandez, E. B., Hybertson, D., Buschmann, F. and Sommerlad, P. (2006). *Security Patterns: Integrating Security and Systems Engineering* (Wiley ed.).
20. Steel, C., Nagappan, R. and Lai, R. (2005). *Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management* (Prentice Hall ed.).
21. Yoder, J. and Barcalow, J. (1997). Architectural Patterns for Enabling Application Security. *Fourth Conference on Patterns Languages of Programs (PLoP'97)*.