SECURITY RISK ANALYSIS IN WEB SERVICES SYSTEMS

Carlos Gutiérrez, Eduardo Fernández-Medina, Mario Piattini
ALARCOS Research Group. Information Systems and Technologies Department UCLM-Soluziona Research and Development Institute. University of Castilla-La Mancha Paseo de la Universidad, 4 – 13071 Ciudad Real, Spain

Keywords: Security Risk Analysis and Management, Security Engineering, Software Security Development Process, Web Services Security.

Abstract: Nowadays, best practices dictate that security requirements of distributed software-intensive systems should be based on security risk assessments. Web services-based systems supporting network alliances among organizations through Internet are such type of systems. In this article we present how we've adopted the risk analysis and management methodology of the Spanish Public Administration, which conforms to ISO 15408 Common Criteria Framework (CCF), to the Process for Web Services Security (PWSSec) developed by the authors. In addition, a real case study where this adaptation was applied is shown.

1 INTRODUCTION

Nowadays, best practices dictate that security requirements of software-intensive systems should be based on risk assessments (Butler and Fischbeck 2005). Software systems based on Web services (WS) technologies have achieved a great popularity recently in both industry and academic world. Web services are a natural consequence of the evolution of the Web and distributed systems. Since its beginnings as a way to share and distribute information on a global scale, effectively becoming a giant distributed content library, the Web has been progressively widening its reach to enable more sophisticated forms of interaction between browser clients and servers: single form-based interactions, retail ecommerce applications, and more complex business-to-business interactions. IDC estimates that $2.3 billion was spent worldwide on total WS software in 2004, more than double the amount from the previous year. IDC expects spending to continue to increase dramatically over the next 5 years, reaching approximately $14.9 billion by 2009. In consequence, security in WS development processes should include a risk analysis so that security requirements can be elicited and prioritized. In this paper, we present a risk analysis process on a WS-based system that is part of the tasks to be developed during the WSSecReq (Web Services Security Requirements) subprocess of the PWSSec (Process for Web Services Security) process created by the authors (Gutiérrez, Fernández-Medina et al. 2005). Although WSSecReq subprocess does not demand a specific risk analysis method we show how the risk analysis and management method of the Spanish Public Administration, Magerit2 (Crespo, Gómez et al. 2005), is applied to a real case study. MAGERIT 2 is the Spanish Public Administration's adaptation of ISO 15408, Common Criteria Framework.

The rest of the article is organized as follows: i) in section 2, a little background on those terms the rest of the article is based on is presented. That is, a brief explanation about the PWSSec process, a short introduction on its WSSecReq subprocess, and, finally, a short presentation of the case study that section 3 is based on (see (Gutiérrez, Fernández-Medina et al. 2005)) for more details on the case study's context); ii) in section 3, we will explain how we have adopted Magerit2 methodology when performing the tasks related to risk analysis defined by the WSSecReq subprocess; iii) in section 4, final conclusions are stated.

Gutiérrez C., Fernández-Medina E. and Piattini M. (2006).
SECURITY RISK ANALYSIS IN WEB SERVICES SYSTEMS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 425-430
DOI: 10.5220/0002105004250430

PWSSec Process
Sub-process P1 – **WSSecReq**
Activity A1.1: Elicitation
Task T1.1.1: Decide granularity level and identify the fragment of functional software whose security will be analyzed
Task T1.1.2: Identify the IBM WS-based business pattern.
Task T1.1.3: Identify the IBM WS-based application pattern.
Task T1.1.4: Identify possible business threats.
Task T1.1.5: Identify possible application threats.
Task T1.1.6: Relate business and application threats.
Task T1.1.7: Identify and assess threats.
Task T1.1.8: Identify type of attackers and their possible types of attack.
Task T1.1.9: Assess impact of attacks.
Task T1.1.10: Estimate and prioritize security risks.
Task T1.1.11: Determine the behaviour the system should have for each attack.
Task T1.1.12: Identify security sub-factors.
Task T1.1.13: Specify security requirements.
Activity A1.2: Analysis
…
Activity A1.3: Specification
…
Activity A1.4: Verification and Validation

Figure 1: Activities and taks of the WSSecReq subprocess.

2 BACKGROUND

2.1 PWSSec Overview

The PWSSec process specifies how to define security requirements for WS-based systems, describes a security services-based reference security architecture and explains how to instantiate it to obtain concrete security architecture based on the current WS security standards (Gutiérrez, Fernández-Medina et al. 2005). PWSSec process is structured in three sub-processes which describe their inputs, outputs, activities, actors and sometimes, guides, best practices, tools and techniques that complement, improve and facilitate the set of activities and tasks developed within these stages. WSSecReq sub-process's main purpose is to produce, by means of a systematic approach, a specification (or a part of it) of the security requirements of the WS-based system. WSSecArch sub-process is aimed at allocating the security requirements specified in the previous section to a WS-based security architecture. This security architecture will be equipped with the necessary security policies and architectural mechanisms to achieve the considered security requirements. WSSecTech subprocess's main objective is to identify the set of WS-based security standards that will implement the architectural security mechanisms identified in the previous stage.

2.2 WSSecReq Overview

The main purpose of this subprocess is to produce a specification (or a part of it) of the security requirements of the target WS-based system. Its input is composed by a specification of the scope that we want to comprise during the current iteration, the business and security goals defined for the system as well as the part of the organizational security policy that we estimate that may impact on the system design. The output is basically formed by: i) A threat attack tree (Schneier 1999) associated with the WS business and application pattern (Endrei, Ang et al. 2004) identified within the analyzed functionality; ii) Every built attack tree's leaf will show a threat (WS-I 2005) that can refined by a set of attack scenarios, defined as misuse cases according to (Alexander 2003; Sindre and Opdahl 2005), organized into attack profiles (Moore, Ellison et al. 2001), and represented according to the Quality of Service UML profile (OMG 2004); ii) every misuse case must have related a set of security use cases, according to Donald G. Firesmith (Firesmith 2003), that state how the system should respond to the associated misuse case; iii) A formal specification of the security requirements for the scope of the system based on SIREN (Toval, Nicolás et al. 2001) (Gutiérrez, Fernández-Medina et al. 2005). These requirements will have been derived after instantiating the WS security requirements templates associated with every security use case. This subprocess defines 4 main activities: **Elicitiation**, **Analysis**, **Specification** and

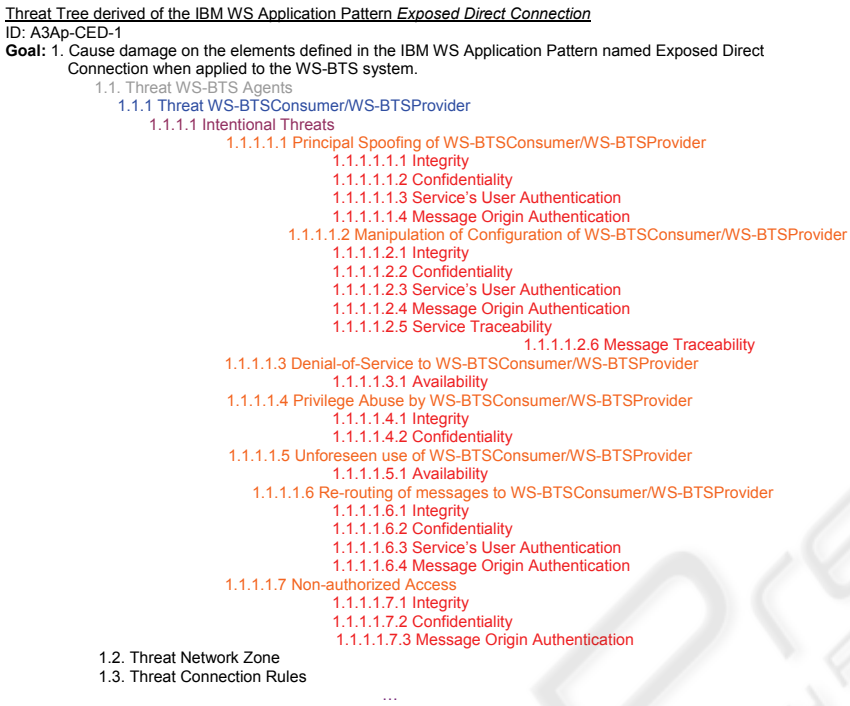

Figure 2: Threat Tree derived from the IBM WS Application Pattern Exposed Direct Connection– View of threats on the existing run-time software systems.

Validation and Verification. Here, we will focus on the **Elicitation** activity (see (Gutiérrez, Fernández-Medina et al. 2005) for more details on the others). The **Elicitation** activity will be supported by a detailed study of security for each WS business service identified and considered in the current iteration. This activity is inspired in the risk analysis and management process known as Operationally Critical Attack, Asset, and Vulnerability Evaluation ^{SM} (OCTAVE) (Firesmith 2003). This activity defines a set of tasks that support security risk analysis during elicitation of security requirements. In this article we will show how we have adopted Magerit2, a Common Criteria Framework-compliant security risk analysis and management methodology, developed by the Spanish Public Administration.

2.3 Case Study

In this article we present an actual case study that was applied to a web services-based system known as WS-BTS (Web Services-based Bank Transfer System). This system's objective was the sale of certain products chosen by purchasers through a Web application. Payments are made from purchaser's bank account which is associated with the bank account of the sales organization (hereafter SalesOrg). This use case was developed as a WS-based system and consists of two types of WS-based agents: (1) a WS consumer agent, belonging to SalesOrg, who will be referred to as WS-BTSConsumer (Web services-based Bank Transfer System Consumer) and (2) the WS provider agent of the bank service (hereafter BankOrg) that will be referred to as WS-BTSProvider (Web services-based Bank Transfer System Provider). These agents interact in order to fulfil a business workflow called BTS (Bank Transfer System), whose objective is to assist the final customer during its payment so purchase is facilitated. This use case is achieved by a three-step protocol carried out by the WS-BTSConsumer and WS-BTSProvider web services agents as described in (Gutiérrez, Fernández-Medina et al. 2005). In this article we illustrate how risk analysis was made as part of applying the WSSecReq subprocess on this case study.

3 RISK ANALYSIS IN WS-BASED SYSTEMS

In this section, we'll show how WSSecReq's tasks were carried out during the aforementioned case study. Concretely, we'll focus on risk analysis-related tasks. That is, tasks from T1.1.4 to T1.1.10 (see high-lighted tasks in Figure 1). In tasks T1.1.1-

T1.1.3, business and application IBM WS-based architectural patterns were identified (Endrei, Ang et al. 2004). The novelty of our approach resides in showing how a risk analysis method conformed to the Common Criteria Framework was integrated into PWSSec in such a way that security requirements and security engineering disciplines for Web services-based system were successfully aligned, integrated and developed. Few previous approaches have been proposed on the subject of applying security risk analysis in WS-based development processes up until now. The problem with them is that they explain how this subject from a very abstract level of detail (Christopher Steel 2005). In this paper, we provide a reusable, real and practical solution on this area showing how we adjusted Magerit2 to security analysis-related tasks of PWSSec.

3.1 A1.1. Elicitation - T1.1.4: Identify Possible Business Threats

Rigorous risk analysis relies on an understanding of business impacts, which requires an understanding of laws and regulations as well as the business model supported by the software (Verdon and McGraw 2004). The main purpose of this task is, from the business-level description elaborated during task T1.1.2, to define the set of potential business-level threats that applies to the system under development. We've associated an abstract business threat tree to every IBM WS business (Endrei, Ang et al. 2004; Gutiérrez, Fernández-Medina et al. 2005). This way, once the WS business pattern has been identified its potential threats are systematically discovered. These threats are organized in a tree-like form (Moore, Ellison et al. 2001). This task's output is a Business Threat Model containing the description of the identified threats organized in the business threat tree. The chosen notational language representation is based on the Quality-of-Service UML Profile (OMG 2004).

3.2 A1.1. Elicitation - T1.1.5: Identify Possible Application Threats

Risk analysis on modern distributed paradigms such as WS, requires a functional decomposition of the application into major components, processes, data stores, and data communication flows, mapped against the environment across which the software will be deployed (Verdon and McGraw 2004). In this task, the application-level threat tree, which provides such a functional decomposition, will be created based on the IBM WS-based application pattern identified during task T1.1.3 (see Figure 2). The set of IBM WS application patterns and their associated abstract threat trees are part of the WS Security E&A (Elicitation and Analysis) Resources Repository of WSSecReq subprocess (Gutiérrez, Fernández-Medina et al. 2005). In Figure 2, the fragment of the application threat tree that unfolds branch 1.1 is presented. Under this branch, the set of threats to be considered on WS agents that participate in the WS-BTS system: Agent WS-BTSConsumer (WS-BTSC) and agent WS-BTSProvider (WS-BTSP) are organized according to their types. The set of threats on the network organized under branch 1.2 and 1.3 are omitted due to space-limits. These threats have been extracted from the catalogue of threats defined in Magerit2. Under branch 1.4 the set of threats to be considered on the WS-based interactions is presented. Here, the division proposed by the abstract threat tree is based on the set of threats on the messages of each one of the interactions that support the functionality whose security is under analysis (threats have been extracted from (WS-I 2005) and (Crespo, Gómez et al. 2005)). This task's output is an Application Threat Model. The description of these threats will give place to a threat model at the application level that will mainly contain: i) An application threat tree specific for the system under analysis; ii) UML QoS model of threats and assets (OMG 2004).

3.3 A1.1. Elicitation - T1.1.7: Threat Assessment

Task T1.1.7 of WSSecReq is completed by applying the following Magerit2's steps: i) **Identification of Assets:** According to the application threat tree, and just focusing on threats on the interactions, the lowest level assets (those whose risk depends on higher-level assets) are TNT message (for the developed branch), TTR Message, TTR Response Message, RNP Message and RNP Response Message as well as WS-BTSP and WS-BTSC agents; ii) **Definition of the Dependency Matrix of Assets:** Every (business/application) abstract threat tree has predefined its own template for its corresponding asset dependency matrix within the *WS Security E&A Resources* WSSecReq's repository. The asset dependency matrix allows the establishment of dependencies between branches representing assets of the threat tree. The types of assets considered in a WS context are: a) **Web**

Table 1: View of the Risk Map showing degradation ratio, accumulated impact and risk of the WS-BTSC asset. Column F represents Frequency of the threat.

			Security Dimensions (I=Integrity, C=Confidentiality, A=Availability, S_A=Service's User Authentication, M_A=Message Origin Authentication, S_T=Service Traceability, M_T=Message Traceability)						
Asset	**Threat**	**F**	**I**	**C**	**D**	**A_S**	**A_M**	**T_S**	**T_D**
WS-BTSC	1.1.1.1.1.1	5	50% [3] {3}	50% [4] {4}		100%[4] {4}	100% [6] {6}		
	1.1.1.1.1.2	5	60 % [4] {4}	5% [0] {0}		10% [0] {0}	10% [0] {0}	0% [0] {0}	0% [0] {0}
	1.1.1.1.1.3	5			10%[0] {0}				
	1.1.1.1.1.4	5	0 [0] {0}	0% [0] {0}					
	1.1.1.1.1.5	5			0%[0] {0}				
	1.1.1.1.1.6	5	10% [0] {0}	5% [0] {0}		5% [0] {0}	5% [0] {0}		
	1.1.1.1.1.7	5	0						
	1.1.1.1.1.8	5	100%[7] {7}	10% [0] {0}		60% [3] {3}			

Services: The purpose of the WS-BTS system is to offer a service; b) **WS agent**: From Magerit2's viewpoint, we consider it as software applications; c) **Messages**: access to data (messages) is made through WS agents; d) **Volatile/Persistent Structured Storage Services** (Databases, directory services, etc.): It is the base from which certain messages are created (outgoing messages) and where the results of processing other messages are stored (incoming messages); iii) **Threat Characterization:** Threat characterization consists of determining the likelier threats for each one of the assets and represents them in a System's Risk Map. In our case, this step was straightforward since we just needed to add two new metrics to the application threat tree: Frequency of Threat Occurrence and Asset Degradation Ratio. The Frequency of Occurrence Threat's value will be valued during task T1.1.8, when all types of attacks for each threat are identified and when the highest frequency of occurrence due to those attacks is obtained. The asset degradation's value will be determined during task T1.1.9 as part of the calculation of the threat impact. In Table 1, the final Risk Map (resulting of task T1.1.10) which includes the set of identified assets is presented. As output product of this task the Threat Assessment, an Assessed Global Threat Model consisting of the aggregation of the security analysis made to the Global Threat Model is obtained.

3.4 A1.1. Elicitation - T1.1.8: Identify the Type of Attackers and their Possible Types of Attacks

The next step will consist of refining the leaf-nodes of the threat tree, i.e. further specification of the threats by means of concrete attacks. Towards this ends, use will be made of the concept of attack profile described in (Moore, Ellison et al. 2001). We use misuse cases in (Sindre and Opdahl 2005) to defining the sequences of steps which state the achievement of successful attacks on the system. An attack profile contains a set of abstract misuse cases that apply to a reference model defined within the profile (in our case the IBM WS-based Application Pattern). Therefore, interactions in every WS-based application pattern have one attack profile related. Every WS-based application pattern has one or more attack profiles related to it which state the potential attacks that could be targeted at them.

We complete the Assessed Global Model of Threats with the characterization and frequency of the attacks that materialize every threat thereby obtaining the Global Model of Threats and Attacks.

3.5 A1.1. Elicitation - T1.1.9: Assess Impact of Attacks

In Magerit2 terms, this task will consist of completing the Risk Map by assigning the value of degradation on assets as a consequence of threats' materialization. In addition, the Risk Map is completed by incorporating an additional value that represents the accumulated impact on every high-level asset (WS-BTSProvider/WS-BTSConsumer) and the repercussed impact on every low-level asset (WS messages). As output of this task we obtain the Assessed Global Model of Threats and Attacks completed with the Risk Map.

3.6 A1.1. Elicitation - T1.1.10: Assess and Prioritize Security Risks

Finally, we estimate and prioritize the risk completing the Assessed Global Model of Threats and Attacks. In the case of Magerit2, risk is computed as a function of the impact and frequency of the threats. Table 1 shows the computed risks for every threat and asset and its security dimension. These risks will guide and provide a basis for the development of the following tasks defined within the WSSecReq sub-process. These tasks basically consist of identifying the expected behaviour of the system for every attack (task T1.1.11) and eliciting the security requirement (task T1.1.12). Risks on every asset will guide what and how resources should be planned during security architecture development (in WSSecArch sub-process).

4 CONCLUSIONS

In this paper, we have presented how Magerit2 can be adapted in the context of the PWSSec process during elicitation of security requirements within WS-based systems. This presentation has been complemented with a demonstration of the application of the WSSecReq subprocess, one of the sub-processess defined by the PWSSec process to a real case study.

ACKNOWLEDGMENTS

This research is part of the following projects RETISTIC network (TIC2002-12487-E), of Dirección General de Investigación del Ministerio de Ciencia y Tecnología, DIMENSIONS (PBC-05-012-1), financed by the FEDER and the “Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha” (Spain) and CALIPO (TIC2003-07804-C05-03) granted by the “Dirección General de nvestigación del Ministerio de Ciencia y Tecnología” (Spain).

REFERENCES

Alexander, I. (2003). "Misuse Cases: Use Cases with Hostile Intent." IEEE Computer Software **20**(1): 58-66.

Butler, S. A. and P. Fischbeck (2005). Multi-Attribute Risk Assessment. SREIS'05 in conjuntction with RE'05, Paris, France.

Christopher Steel, R. N., Ray Lai (2005). Core Security Patterns: Best Practices and Strategies for J2EE™, Web Services, and Identity Management, Prentice Hall PTR / Sun Microsystems.

Crespo, F. L., M. Á. A. Gómez, et al. (2005). MAGERIT - Versión 2. Metodologías de Análisis y Gestión de Riesgos de los Sistemas de Información. III - Guía de Técnicas. Madrid, Ministerio de Administraciones Públicas**:** 154.

Endrei, M., J. Ang, et al. (2004). Patterns: Service-Oriented Architecture and Web Services**:** 345.

Firesmith, D. G. (2003). "Engineering Security Requirements." Journal of Object Technology **2**(1): 53-68.

Firesmith, D. G. (2003). "Security Use Cases." Journal of Object Technology **2**(3): 53-64.

Gutiérrez, C., E. Fernández-Medina, et al. (2005). PWSSec: Process for Web Services Security. IEEE International Conference on Web Services 2005, Orlando, Florida, USA.

Gutiérrez, C., E. Fernández-Medina, et al. (2005). Security Requirements for Web Services based on SIREN. Symposium on Requirements Engineering for Information Security, Paris, France.

Gutiérrez, C., E. Fernández-Medina, et al. (2005). Web Services Enterprise Security Architecture: a Case Study. ACM Workshop on Security on Web Services, Fairfax, Virginia, USA, ACM Press.

Gutiérrez, C., E. Fernández-Medina, et al. (2005). Web Services-based Security Requirement Elicitation. 1st International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements (SOCCER'05) in conjunction with IEEE RE'05, Paris, France.

Moore, A. P., R. J. Ellison, et al. (2001). Attack Modelling for Information Security and Survivability. Survivable Systems, Software Engineering Institute.

OMG (2004). UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms.

Schneier, B. (1999). "Attack Trees: Modeling Security Threats." Dr. Dobb's Journal.

Sindre, G. and A. L. Opdahl (2005). "Eliciting Security Requirements with Misuse Cases." Requirements Engineering Journal **10**(1): 34-44.

Toval, A., J. Nicolás, et al. (2001). "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach." Requirements Engineering Journal **6**(4): 205-219.

Verdon, D. and G. McGraw (2004). Risk Analysis in Software Design. IEEE Security & Privacy. **2:** 79-84.

WS-I (2005). Security Challenges, Threats and Countermeasures Versión 1.0, WS-I. **2005**.