Towards a Process for Web Services Security

Carlos Gutiérrez

STL, Madrid (SPAIN)
carlos.gutierrez@stl.es

Eduardo Fernández-Medina and Mario Piattini

Alarcos Research Group, Universidad de Castilla-La Mancha,
Paseo de la Universidad 4, 13071, Ciudad Real. (SPAIN). Tel: 34 926 29 53 00
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Web Services (WS) security has undergone an enormous development, as carried out by the major organizations and consortiums of the industry over the last few years. This has brought about the appearance of a huge number of WS security standards. Such a fact has made organizations remain reticent about adopting technologies based on this paradigm, due to the learning curve which is inevitable in the integration of security into their practical deployments. In this paper we present PWSSec (Process for Web Services Security), which enables the integration of a set of specific security stages into the traditional phases of WS-based systems development. PWSSec defines three stages, WSSecReq (Web Services Security Requirements), WSSecArch (Web Services Security Architecture) and WSSecTech (Web Services Security Technologies). These facilitate, respectively, the definition of WS-specific security requirements, the development of a WS-based security architecture and the identification of the WS security standards that the security architecture must articulate in order to implement the security services.

ACM Classification: D.2.1 (Requirements/Specification)s, D.2.11 (Software Architecture), D.2.12 (Interoperability), D.2.13 (Reusable Software)

1. INTRODUCTION

The open nature of the Internet is promoting the development of Web Services (WS) as a paradigm that enables complex scenarios of business workflow integration, while at the same time providing the intra and inter-firm hyper-connectivity that is so much of a current buzz-term and in such wide demand (Nott, 2004).

This standard-based quality-centred paradigm is evolving rapidly due to its capability in handling and addressing the challenge of heterogeneous system integration. In fact, according to the most recent reports from IDC, over the next years the market for WS-based solutions will grow steadily, reaching $11 billion in 2008 (IDC, 2004). Consequently, an enormous quantity of WS standards is being produced. This diversity, also found in the context of WS (Gutiérrez *et al*, 2004) security has made us see its application, as being a very complex process, considered overall. It is hard to understand, implying a very steep learning curve. At this point, we can set out some questions that are hard to answer for someone who is new to WS-based systems:

Manuscript received: 12 April 2005
Communicating Editor: Julio Cesar Hernandez

- Given a complete set of functional requirements, what WS security standards should I choose?
- What are the WS security requirements that should be taken into account in my WS-based system?
- Which WS security standard from among those addressing similar aspects should I integrate in my WS-based system?

At present, there is still a lack of a comprehensive approach which offers a methodical development in the construction of security architectures for WS-based systems. Thus, the main objective of this paper is to address this need, presenting the process named PWSSec (Process for Web Services Security). PWSSec has been created to facilitate and orientate the development of security for WS-based systems in such a way that in each one of the traditional stages for the development of this kind of systems (Endrei *et al*, 2004), a complementary stage including security can be integrated. Therefore, this process can be used once the functional architecture of the system has been built, or during the stages used to produce this architecture. In both cases, the result will be a WS-based security architecture formed by a set of coordinated security mechanisms using the WS security standards to fulfill the WS-based system security requirements.

In this article we will provide a brief overview of the complete process and present, in greater detail, the reference security architecture as defined in the WSSecArch stage.

The rest of the paper is organized as follows. In Section 2, the PWSSec process is introduced. In Section 3, an in-depth study of the stage related to the specification of the WS-based security architecture is presented and an example is shown. In Section 4, related research works are outlined, and, finally, in Section 5 conclusions and issues that need to be developed in the future are enumerated.

2. PWSSEC – PROCESS FOR WEB SERVICES SECURITY

In this section, we will provide an overview of the process PWSSec.

This process specifies how to define security requirements for WS-based systems, describes a WS reference security architecture that guarantees and demonstrates its development and provides us with facilities for obtaining specific security architectures based on the current WS security standards. In general, the main features of the process presented in this section are as stated below.

Iterative and incremental process: for each iteration that comprises the development of all stages, a part (increase) of system security is analyzed, characterized and developed (Breu *et al*, 2003).

- The two basic principles are process traceability and reusability and product interoperability (WS-I, 2004) and reusability. Process reusability will allow us to apply it to different problem domains in which it is necessary to develop a WS-based security architecture, whilst product reusability will guarantee shorter development cycles based on proven solutions. Product interoperability, mainly applied in WSSecArch and WSSecTech stages, will guarantee that WS-based security solutions agreed on by the most important industry consortiums will be taken so that systems developed with PWSSec will present a high degree of integration and interoperability.
- The process is managed by the elements and basic procedures defined for an Architecture based on WS (Papazoglou and Georgakopoulo, 2003): the basic actors are the services provider agents, the services consumer agents and the discovery agents, whilst the basic processes are publishing, discovery, binding and invocation.
- It is based on the concept and techniques developed within the scope of Security Requirement Engineering and Risk Analysis and Management (Alberts *et al*, 1999; Smith, 2003; OMG, 2004).

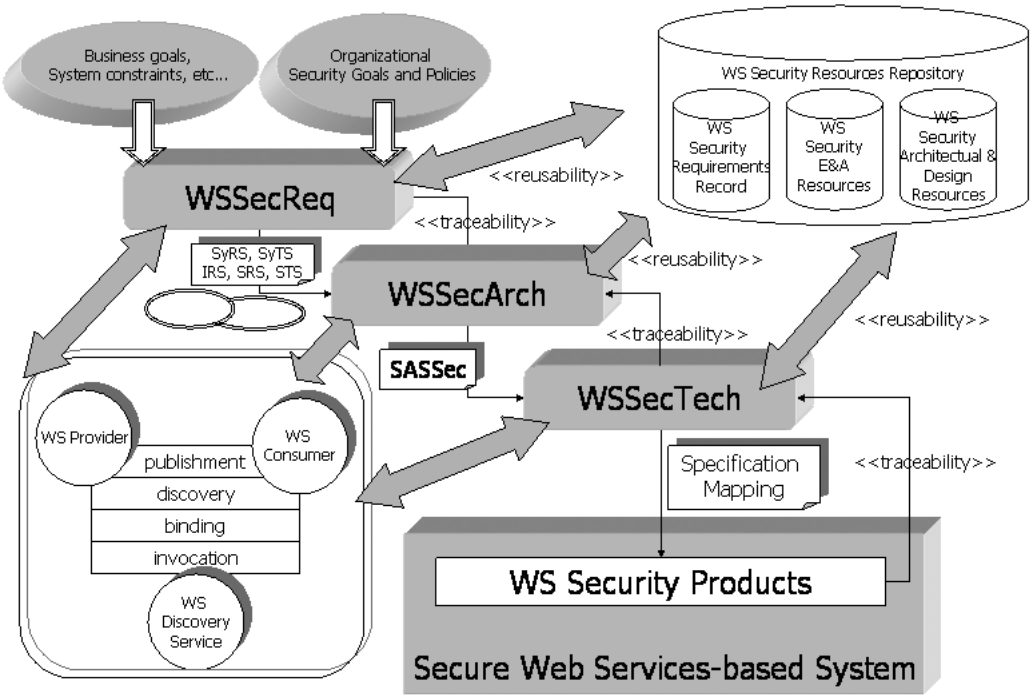

Figure 1: PWSSec process layout

- It is developed from the concept and techniques that allow us to implement security into software architecture (Bass and Kazman,1999; Jürjens, 2005; Yu *et al*, 2004).

Figure 1 shows us the stages in which PWSSec process is structured.

Each one of the stages defined by PWSSec describes its inputs, outputs, activities, actors and, in various cases, some guides, good practices, tools and techniques all serving to complement, improve and facilitate the set of activities developed within these stages.

The PWSSec stages depicted in Figure 1 are described briefly as follows:

- WSSecReq (Web Services Security Requirements): The main purpose of this stage is to produce a specification (or a part of it) of the security requirements of the target WS-based system. Its input is made up of a specification of the scope that we want to include during iteration (e.g.: if we have a Use Cases Model available, we can select those uses cases that we want to cover and use them as an input for iteration) and the business and security goals defined for the system, as well as the part of the organizational security policy that we estimate can impact on the system design. The output is basically formed by the set of attack scenarios, defined according to (Sindre and Opdahl, 2000; Alexander, 2003) and represented according to the UML profile (OMG, 2004), by the set of use cases of security according to Donald G. Firesmith (Firesmith, 2003) and by a formal specification of the security requirement for the scope of the system based on SIREN (Toval *et al*, 2001). This stage is supported by a repository that contains attack scenario patterns which are grouped into attack profiles and security use cases, by a set of reusable security requirement templates and by a basic guide for the definition of scenarios and security requirements within WS-based systems.

- WSSecArch (Web Services Security Architecture) has as its main objective to allocate and integrate the security requirements specified in the WSSecReq stage, through the identification of the appropriate WS-based security architectural patterns and their integration in a WS-based security architecture. We skip a brief description of this stage because we give a detailed development in Section 3.
- WSSecTech (Web Service s Security Technologies): The main purpose of this stage is to identify a set of WS security standards that will implement the security services identified in the previous stage. Output will be a description of the set of standards identified for each security service, together with the reasoning framework that made us select it and a specific security architecture design. The activities carried out in this stage are the following: i) WS Security Standards Identification; ii) Security Policies Specification to define the Security Policy of the Abstract Security Service Instance that implements a certain standard.

3. WSSECARCH – WEB SERVICES SECURITY ARCHITECTURE

WSSecArch comprises the security-related Architecture Design which provides a clear distribution of the security requirements into WS-based security mechanisms. These WS-based security mechanisms help to mitigate those risks identified for every business Web Service in the WSSecReq stage and complete the WS-based security architecture.

The two principles that characterize this stage are the same as in WSSecReq: product reusability and traceability. Architecture security solutions are taken from architectural patterns (Bass *et al*, 2003) that have been successfully proved in other projects and which belong to the most diverse domains. Thus, a repository of WS security architectural patterns related to each one of the security factors, has been created, including the reasoning framework that relates and justifies them (Bass *et al*, 2004; Klein and Kazman, 1999). Nowadays, there are different WS-based security solutions for the different security requirements that we can identify. Up to this point we have focused our work on federation environments including authentication, Single Sign-On and domain trust federation topics.

The considered inputs in this stage are:

- Business goals of the current iteration.
- Organizational goals and security policies taken into account during the current iteration.
- The set of attack and security scenarios developed in WSSecReq.
- The set of security requirements specifications defined in WSSecReq stage.
- The WS Security Architectural & Design Repository which contains a set of WS-based architectural security patterns that will promote product reusability.

The main output of this stage is the Security Architecture Specification (SASSec in Figure 1) which accomplishes:

- The set of security requirements solved by the architecture.
- The set of security architectural patterns that implement them, documented as in Attribute-Based Architectural Styles (Klein and Kazman, 1999).
- The catalog of security policies associated with the business WS and security WS.
- A series of views (Bass *et al*, 2003; Krutchen, 1995) whose type depends on the stakeholders that will read the SASSec. These views will demonstrate how the security architecture integrates with the functional architecture and how the attack scenarios are addressed. Between the security requirements, the identified security patterns and the WSSecKern (Web Services

Security Kernel) components (we will get on to this concept later), a forward and backward navigable traceability relationship, justified by the set of design decisions or applied reasoning, will exist.

The main actors in this stage are the Requirements Engineering Team, the Architecture Design Team and the Security Team. The last two participate in the Security Architectural Patterns Identification and Security Architectural Patterns Integration stages. In the Security Architecture Validation stage the three actors participate.

3. 1 Activities

3. 1. 1 Security Architectural Patterns Identification

For each security requirement of every business WS belonging to the current iteration, we must identify the WS security architectural pattern(s) that solve(s) it. This architectural pattern defines a set of abstract security-related service types (since they do not define how they must be implemented in terms of specific WS security standards) as well as a set of interactions that formally specify the security properties offered by the pattern. The WS architectural pattern, defined as a set of coordinated design (Bass *et al*, 2003), adds a set of additional security services to which we must add a description of their interaction (interaction between security services and between security services and business WS) in the functional service architecture. Therefore, these new services will offer us new security functionalities that must be reevaluated, analyzed and refined from the WSSecReq process.

As a complementary source for this stage, the WS Security Architectural & Design Repository (see Figure 1) should be used. New identified architectural patterns will be introduced within the repository, allowing its future reuse.

3.1.2 Definition of Security Policy associated with every Security Architectural Pattern

Furthermore, such an architectural pattern defines, in an abstract way, the possible security policy parameters that can govern the capabilities and interactions of the identified security mechanisms. The security policies allow Abstract Security Services and business WS to define their preferences, requirements and capabilities (VeriSign *et al*, 2004; Anderson *et al*, 2004). Each Abstract Security Service, derived from one or more security requirements through the application of a certain architectural pattern(s), must indicate in its security policy the possible parameters for instantiating it and the set of security requirement types it addresses (e.g.: authentication, availability, etc.).

3.1.3 Integration of Security Architectural Patterns

With the purpose of obtaining a systematic method to be able to define the WS-based architecture security, we have elaborated a WS-based security reference architecture that shows the direct traceability of security requirements with their corresponding components of implementation software. In our work we have developed a WS security reference architecture which is shown in Figure 2.

The basic elements used in the WS security reference architecture are:

- WS Security Kernel (WSSecKern). This is the core of our WS-based reference security architecture. This component will manage a set of Abstract Security Services, derived from the application of a certain set of security architectural patterns, with the aim of supporting the security requirements of a potential set of business Web Services. Each WSSecKern will support one or more Abstract Security Services, implemented through one or more specific security mechanisms in the form of WS security standards (identified in the stage WSSecTech). Thus,

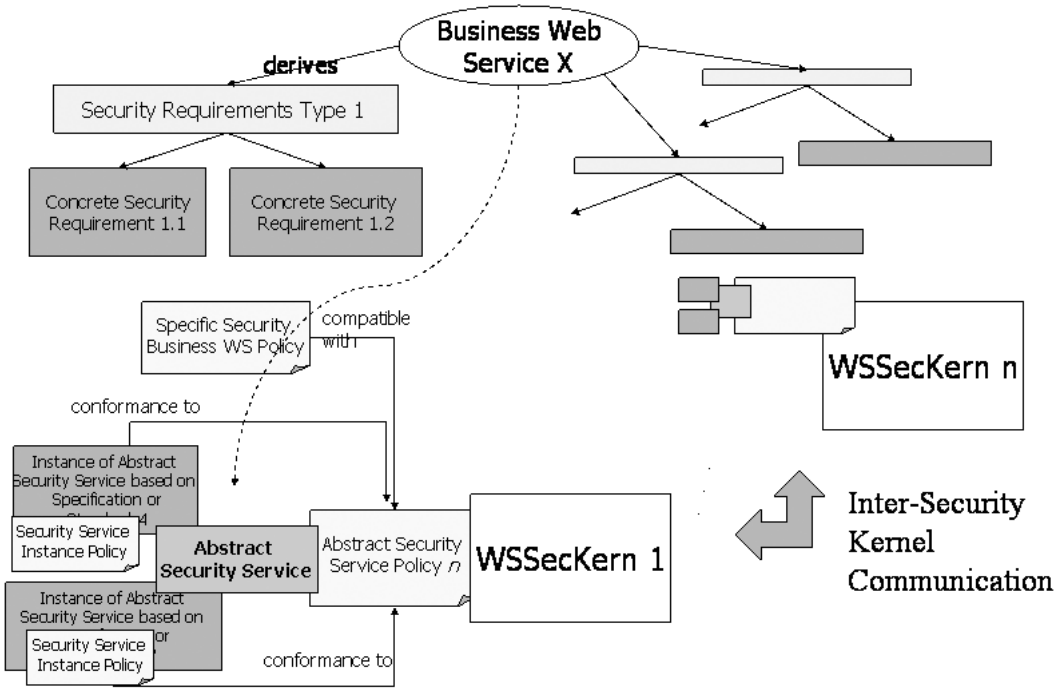

Figure 2: WS Abstract Security Reference Architecture

each WSSecKern will cover a set of security requirements by means of providing a set of WS-based security services to certain business WS.

- Abstract Security Service, comprising a certain set of security requirement types (e.g.: security requirements related to authorization) and which can have several instances, according to the number of implementations based on the WS security standards that will be identified in the stage WSSecTech.
- Security Policy of an Abstract Security: this includes the possible parameters or attributes with which we can define the security policies of potential instances of the Abstract Security Service as well as a description of the set of security requirement types that the Abstract Security Service handles.
- WS Security Standard, which supports a certain Abstract Security Service Instance (indeed, this will not be defined until the WSSecTech stage).
- Security Policy of an Abstract Security Service Instance in which the capabilities supported by the specification or standard used are defined.
- Business Service Security Policy: defined by each business WS. The business WS security policy will be registered in the WSSecKern when the business WS wishes to use the security services provided by that WSSecKern. Hence, the WSSecKern will know what security services are demanded by certain WS, and how to use them. The business WS defines what set of security requirements it needs as well as which mechanisms and how these will be used (e.g.: "I would like a simple message authentication based on X.509v3" certificates).
- Protocol of Intercommunication between WSSecKern: It allows the coordination and interaction of the different Security Services.

On the other hand, the basic interactions are:

- Registration/cancellation of the business WS in WSSecKern. A business WS must register itself in a WSSecKern including the definition of its Business Service Security Policy. This way the WSSecKern will know what business WS should protect, what security services will have to be applied and how.
- Execution of an operation of a Security Service Instance. When a request arrives at a business WS, depending on the way the system is configured, this could be intercepted by a certain WSSecKern or it could be forwarded by the business service to a WSSecKern so that the security service may be effectively applied.

3.1.4 Security Architecture Validation

This activity mainly consists of verifying that the attack and security scenarios for the current iteration are covered. Besides this, scenarios that have been identified so far must be reevaluated, with the purpose of checking that a conflict situation has not begun.

3.1.5 Security Architecture Specification

This activity gives a written statement in the form of a Security Architecture Specification document (SASSec in Figure 1) through the use of views (Krutchen, 1995; Bass *et al*, 2003). These show us how the security scenarios display, through the architecture components interactions (WSSecKern and its Abstract Security Services, Agents WS Consumers and Agents WS Providers), as a countermeasure to the attack scenarios shown in the current iteration. Moreover, the specification must show the distribution of the security requirements given as input in a way that each WSSecKern must perform one or more security requirements.

As an example, Figure 3 shows us a specific security architecture covering the authentication requirements and the perimeter security (Cremonini *et al*, 2003):

In Figure 3, we can see how the security requirement "Perimeter security" defined from the business WS analysis can be specified in a Filtering Service whose aim is to filter messages according to some simple syntactic rules of these defined in their policy, and which are directed to that business service. The business service, with respect to this R1 requirement, will indicate, in its security policy, the following points:

- Requires Perimeter Security.
- Requires Message Filtering.
- R1 is an assigned requirement.
- Filtering must be syntactic-based, applied to the message payload and based on a certain XML schema, defined in the business service policy.

In this example, the WSSecKern has been implemented as an Enterprise Security Service Bus (Nott, 2004) specialized in security. The ESBSec must implement the defined semantics for a WSSecKern and has a WSFiltSrv Abstract Security Service associated. This latter is derived from the Message Filtering security requirement discovered when applying WSSecReq to the given business service. WSFiltSrv defines, as parameters of its policy, the definition of the formats in which the syntactic rules can be expressed (e.g.: XML Schema, DTD, RelaxNG, etc...). Instances of this Abstract Security Service must define what formats they support.

Given that there is not any standard or specification related to the XML security filtering topic; WSFiltSrv is based on a custom solution that defines in its Security Policy of the Abstract Security Service Instance its capability to specify the syntactic rules through the XML Schema and XPath

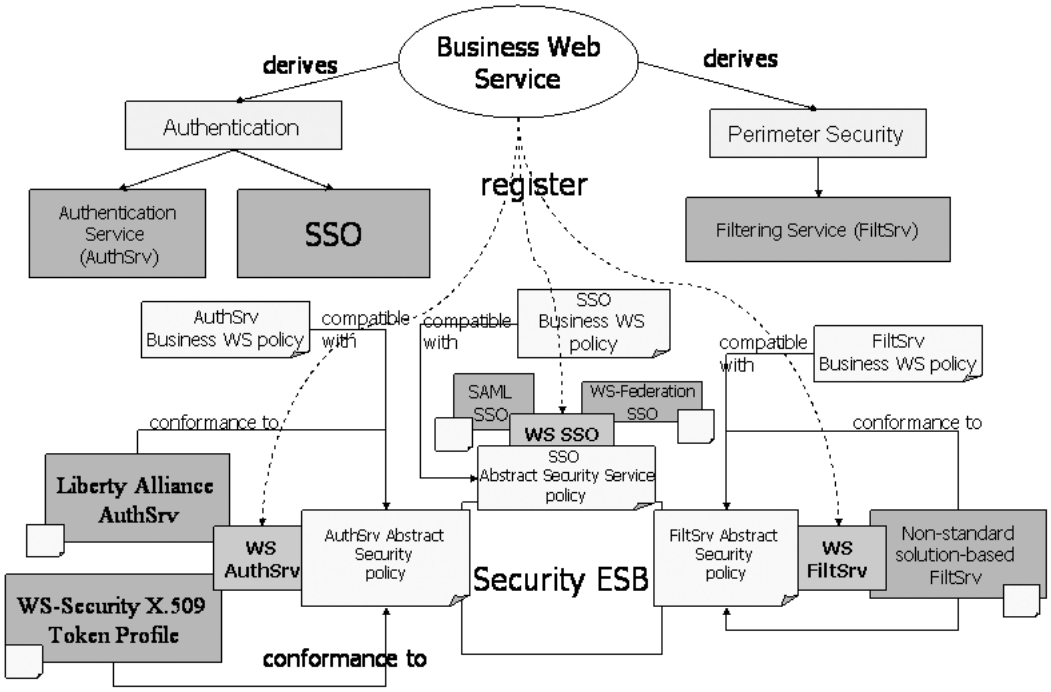

Figure 3: Concrete WS security Architecture

expressions. When the business WS is registered in ESBSec, it will indicate that it requires Perimeter Security of Filtering Message type and it will provide the syntactic rules in XML Schema and XPath expressions format as well. When ESBSec carries out the compatibility verification task, it will note that it has an Abstract Security Service that implements the type of Message Filtering requirements and that also, and having analyzed the Instances security policies (only one, in this particular case), there is at least one instance that accepts the XML Schema and XPath expressions format for filtering syntactic rules. As verification is correct, R1 requirement will be implemented by a WSSecKern in the shape of ESBSec. We must highlight the fact that it is possible to know at runtime which component or (sets of software components) implement(s) a certain security requirement. This fact will provide us with total requirement traceability, from the security requirement to the architectural component that addresses it.

4. RELATED WORK

At present, undoubtedly, the biggest effort is being carried out in the area of definition in WS security standards. This effort has brought into existence a vast number of drafts and standards whose wide range makes it difficult for the organizations that would like to use them, or even to be aware of all those that exist. The lack of a global vision has caused many organizations to be very reticent in the use of this method since they have seen it as being full of acronyms. Our process allows us to face this problem in an orderly way, enabling organizations to apply this method without having to know previously which draft or standard must be put to work.

With regard to this research area, EFSOC (Leune *et al*, 2004) is a event-driven framework for WS-based system development, defining a security model that can be easily applied to systems in

which the modifiability degree is high. As such, therefore, they require a review and update of authorization policies. In (Deubler *et al*, 2004), a methodical and formal analysis based on "formal analysis of security-critical service-based software systems" is presented. None of these approaches puts forward a method such as PWSSec, which, from the business and system security goals, can obtain a system based on secure WS and which is defined upwards, as far as the level of the standards used. Moreover, none of these methods offers us facilities for the reusability of the generated products in such a way that their practical applicability is guaranteed.

5. CONCLUSION AND FUTURE RESEARCH

This paper has presented the process PWSSec, which allows us to provide a WS system with security through a systematic process. As far as the authors of this paper know, in the field of WS system research, there is no existing definition of a complete process which includes and takes into account all the stages of its life cycle. At the present time we are applying PWSSec to two study cases carried out by a couple of state-owned organizations. We hope that, as a result of this practical application, we could refine the stages of the process and enrich the products generated in it.

Some of the main aspects to be developed are the following: to complete the repository defined in the WSSecReq stage with security requirement templates and specific attack patterns that include more security aspects; WS security requirement modelling and formal validation; to develop evaluation areas and cost/benefit analysis of WSSecArch; to complete the catalog of WSSecArch standards and specifications (completed for authentication and perimeter security requirements thus far); in the process of verification of policy compatibilities executed by WSSecKern, we still have to define if two policies are semantically equivalent.

ACKNOWLEDGMENTS

This research is part of the following projects: MESSENGER (PCC-03-003-1) financed by the "Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" (Spain), CALIPO (TIC2003-07804-C05-03) and RETISTIC (TIC2002-12487-E) granted by the "Dirección General de Investigación del Ministerio de Ciencia y Tecnología" (Spain).

REFERENCES

ALBERTS, C. J., BEHRENS, S. G., PETHIA, R. D. and WILSON, W. R. (1999): Operationally critical threat, asset, and vulnerability evaluation (OCTAVE) framework, Version 1.0., Carnegie Mellon, Software Engineering Institute.

ALEXANDER, I. (2003): Misuse cases: Use cases with hostile intent, *IEEE Computer Software*, 20: 58-66.

ANDERSON, A., PROCTOR, S. and GODIK, S. (2004): OASIS XACML profile for Web-services.

BASS, L., BACHMANN, F., ELLISON, R. J., MOORE, A. P. and KLEIN, M. (2004): Security and survivability reasoning frameworks and architectural design tactics, Carnegie Mellon, Software Engineering Institute.

BASS, L., CLEMENTS, P. and KAZMAN, R. (2003): Software architecture in practice, Addison-Wesley.

BASS, L. and KAZMAN, R. (1999): Architecture based development, Carnegie Mellon, Software Engineering Institute, April.

BREU, R., BURGER, K., HAFNER, M., JÜRJENS, J., POPP, G., WIMMEL, G. and LOTZ, V. (2003): Key Issues of a formally based process model for security engineering, in *ICSSEA'03*.

CREMONINI, M., DAMIANI, E., VIMERCATI, S. D. C. D. and SAMARATI, P. (2003): A XML-based approach to combine firewalls and web services security specifications, in *ACM Workshop on XML Security*, 69-78.

DEUBLER, M., GRÜNBAUER, J., JÜRJENS, J. and WIMMEL, G. (2004): Sound development of secure service-based systems, in *ICSOC'04*.

ENDREI, M., ANG, J., ARSANJANI, A., CHUA, S., COMTE, P., KROGDAHL, P., LUO, M. and NEWLING, T. (2004): Patterns: Services oriented architectures and web services.

FIRESMITH, D. G. (2003): Security use cases, *Journal of Object Technology*, 2: 53-64

GUTIÉRREZ, C., FERNÁNDEZ-MEDINA, E. and PIATTINI, M. (2004): Web services security: is the problem solved?, *Information Systems Security*, 13: 22-31

IDC (2004): see http://www.idc.com/getdoc.jsp?containerId=pr2004_04_13_090643.

JÜRJENS, J. (2005): Secure systems development with UML, Springer.
KLEIN, M. and KAZMAN, R. (1999): Attribute-based architectural styles, Carnegie Mellon, Software Engineering Institute.
KRUTCHEN, P. (1995): The 4+1 view model of software architecture. *IEEE Software*, 12(6): 42-50.
LEUNE, K., PAPAZAGLOU, M. and HEUVEL, W.-J. v. d. (2004): Specification and querying of security constraints in the EFSOC framework, in *ICSOC'04*.
MOORE, A. P., ELLISON, R. J. and LINGER, R. C. (2001): Attack modelling for information security and survivability, Carnegie Mellon, Software Engineering Institute.
NOTT, C. (2004): Patterns: Using business service choreography in conjuction with an enterprise service bus.
OMG (2004): UML profile for QoS and fault tolerance, see http://www.omg.org/docs/ptc/04-09-01.pdf
PAPAZOGLOU, M. P. and GEORGAKOPOULO, D. (2003): Service-oriented computing, *Communications of the ACM*, 46 (10): 25-28.
SINDRE, G. and OPDAHL, A. L. (2000): Eliciting security requirements with misuse cases, in *TOOLS-37'00*, 120-131.
SMITH, D. G. (2003): Common concepts underlying safety, security, and survivability engineering, Carnegie Mellon, Software Engineering Institute.
TOVAL, A., NICOLÁS, J., MOROS, B. and GARCÍA, F. (2001): Requirements reuse for improving information systems security: A practitioner's approach, *Requirements Engineering Journal*, 6: 205-219.
VeriSign, MICROSOFT, SonicSoftware, IBM, BEA and SAP (2004): Web services policy framework (WS-Policy), see http://www-128.ibm.com/developerworks/library/specification/ws-polfram/.
WS-I (2004): www.ws-i.org.
YU, H., HE, X., DENG, Y. and MO, L. (2004): Integrating security administration into software architecture design, in *ICSEK'04*.

BIOGRAPHICAL NOTES

Carlos Gutiérrez

Carlos Gutiérrez has an MSc from the Autonomous University of Madrid (Spain) and currently he is a PhD candidate and Assistant Professor at the University of Castilla–La Mancha. He has developed his professional activities in national and international companies doing consultancy work. He is currently an Internet analyst in Sistemas Técnicos de Loterías del Estado (State Lotteries' Technical Systems). His research activities are focused on web services security and secure software architectures. He has published several papers in international conferences and diverse articles in national and international magazines on these subjects. He is participating at the ALARCOS research group and he is an ACM member. His e-mail address is: carlos.gutierrez@stl.es.

Eduardo Fernandez-Medina

Eduardo Fernández-Medina holds a PhD and an MSc in Computer Science from the University of Sevilla. He is Assistant Professor at the Escuela Superior de Informática of the University of Castilla-La Mancha at Ciudad Real (Spain), his research activity being in the field of security in databases, datawarehouses, web services and information systems, and also in security metrics. Fernández-Medina is co-editor of several books and chapter books on these subjects, and has dozens of papers in national and international conferences (DEXA, CAISE, UML, ER, etc.). Author of several manuscripts in national and international journals (Information Software Technology, Computers And Security, Information Systems Security, etc.), he is a member of the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. He belongs to various professional and research associations (ATI, AEC, ISO, IFIP WG11.3 etc.). Eduardo's e-mail is eduardo.fdezmedina@uclm.es.

Mario Piattini has an MSc and a PhD in Computer Science from the Politechnical University of Madrid. He is a Certified Information System Auditor from the ISACA (Information System Audit and Control Association). Full Professor at the Escuela Superior de Informática of the Castilla-La Mancha University (Spain) and author of several books and papers on databases, software engineering and information systems, Piattini leads the ALARCOS research group of the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. His research interests are: advanced database design, database quality, software metrics, object- oriented metrics and software maintenance. His e-mail address is Mario.Piattini@uclm.es.

Mario Piattina