World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:1, 2009

Web Service Security Method To SOA Development

Nafise Fareghzadeh

***Abstract*—** Web services provide significant new benefits for SOA-based applications, but they also expose significant new security risks. There are huge number of WS security standards and processes. At present, there is still a lack of a comprehensive approach which offers a methodical development in the construction of secure WS-based SOA. Thus, the main objective of this paper is to address this needs, presenting a comprehensive method for Web Services Security guaranty in SOA. The proposed method defines three stages, Initial Security Analysis, Architectural Security Guaranty and WS Security Standards Identification. These facilitate, respectively, the definition and analysis of WS-specific security requirements, the development of a WS-based security architecture and the identification of the related WS security standards that the security architecture must articulate in order to implement the security services.

***Keywords*—** Kernel, Repository, Security Standards, WS Security Policy, WS specification.

I. Introduction

The major advance of Web services technologies promises to have far reaching effects on the Internet and enterprise networks. Web services based on the eXtensible Markup Language (XML), SOAP, and related open standards, and deployed in Service Oriented Architectures (SOA) allow data and applications to interact without human intervention through dynamic connections.

The security challenges presented by the Web services approach are formidable and unavoidable and there is still a lack of a comprehensive approach which offers a methodical development in the construction of security architectures for WS-based SOA. The proposed method in this paper has been created to facilitate and orientate the development of security for WS-based SOA in such a way that in each one of the traditional stages for the development of this kind of systems, 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 secure service oriented architecture formed by a set of coordinated security mechanisms using the WS security standards to fulfill the WS-based system security requirements.

The remainder of the paper is organised as follows: In Section 2, the Suggested WS Security Method to develop SOA and it's stages are introduced and discussed. In Section 3, related research works are outlined, and, finally, in Section 4 conclusions and issues that need to be developed in the future are enumerated.

Nafise Fareghzadeh is a is the associate professor at Computer Department of Islamic Azad University of Zanjan and Tehran, Payame Noor University of Tehran, Raja Uniersity of Quazvin (e-mail: n_f2840@yahoo.com).

II. Suggested Method

Security remains one of the core cross-cutting concerns in integrated web service based SOA governance. One of the key roles of any integrated SOA governance solution is to ensure that services are delivered and accessed securely according to enterprise security policies.

The proposed method specifies how to define security requirements for WS-based SOA 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. The method is managed by the elements and basic procedures defined for an Architecture based on WS (Papazoglou and Georgakopoulo, 2003) and 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). It is developed from the concept and techniques that allow us to implement security into software architecture. The two basic principles in suggested method are process traceability and reusability and product interoperability 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 Architectural Security Guaranty and Security Standards Identification stages, will guarantee that WS based security solutions agreed on by the most important industry consortiums will be taken so that systems developed with suggested method in this paper will present a high degree of integration and interoperability [1,4].

The proposed method defines three stages, Initial Security Analysis, Architectural Security Guaranty and WS Security Standards Identification. These facilitate, respectively, the definition and analysis of WS-specific security requirements, the development of a WS-based security architecture and the identification of the related WS security standards that the security architecture must articulate in order to implement the security services. The following picture describes the general structure of the suggested method:

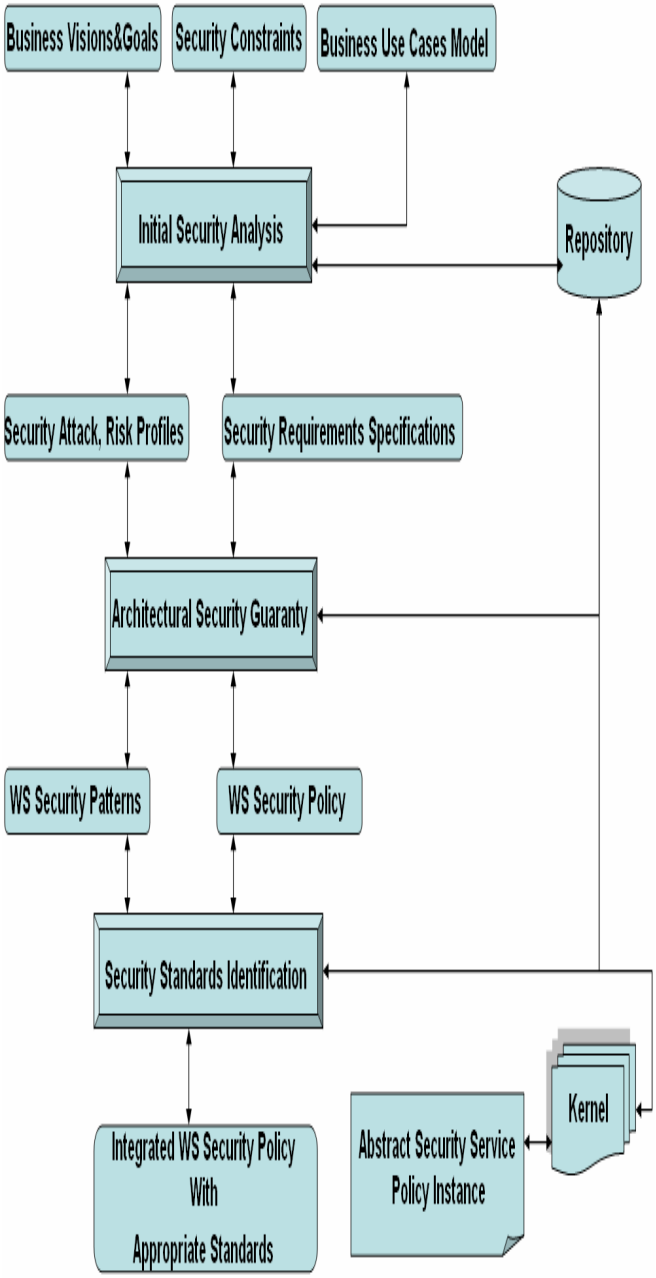

Fig. 1 Web Service Security Method Stages

Suggested method, represented in the last Figure, includes the following stages:

A. Initial Security Analysis

Inputs:

- Project vision, strategies;
- Organizational security goals, scope, requirements;
- Business goals, System constraints, Rules;
- Business Use Cases Model, Use Cases;

The main purpose of this phase is to produce a complete specification of the security requirements of the target secure WS-based system in SOA. During the initial security analysis phase, basic decisions regarding project visions and goals and purpose of the security analysis, security policy scope, security requirements are to be made. The area of security analysis may be documented by a framework, identifying the most important security functions of the domain and showing their requirements. This framework can act as a starting point for the next stages. Its input is made up of a specification of the scope that we want to include during 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. So first we should use Business Use Cases Model and select those uses cases that we want to cover and use them as an input for iteration then we can extract important security goals and requirements. The output is basically formed by the set of attack scenarios, and can represent according to the UML profile, by the set of use cases of security by a formal specification of the security requirement for the scope of the system. 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 [5-7-11].

Outputs:

- Attack scenarios patterns, Security policies;
- Formal specification of the security requirement;
- Security attack, Risk profiles;

B. Architectural Security Guaranty

Inputs:

- Security policies, Business and organizational goals of the current iteration;
- The set of attack and security scenarios developed in Initial Security Analysis stage;
- The set of security requirements specifications;
- The Repository which contains a set of WS-based architectural security patterns;

The main objective in this phase is to allocate and integrate the security requirements specified in the Initial Security Analysis stage, through the identification of the appropriate WS-based security architectural patterns and their integration in a WS-based security architecture. WS security architecture 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 Initial Security Analysis stage and complete the WS-based security architecture. 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).

So, first of all for security requirements specified in the Initial Security Analysis stage, 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 as well as a set of interactions that formally specify the security properties offered by the pattern. As a complementary source for this stage, the WS Security Repository should be used. New identified architectural patterns will be introduced within the repository, allowing its future reuse [8-11].

Furthermore, we must define appropriate Security Policy associated with every Security Architectural Pattern. 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. The security policies allow Abstract Security Services and business WS to define their preferences, requirements and capabilities.

To obtain a systematic method to be able to define the WS-based secure SOA, we have proposed a WS-based security reference architecture that shows the direct traceability of security requirements with their corresponding components of implementation software. In this paper we have developed a WS security reference architecture based on proposed method which is shown in Figure 1. The most important element of the WS security reference architecture is Kernel. 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 Kernel will support one or more Abstract Security Services. Abstract Security Service, comprise 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 (which supports a certain Abstract Security Service Instance) that will be identified in the stage Security Standards Identification. Abstract Security Service also includes Security Policy that 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. In the security reference architecture also Business Service Security Policy defined by each business WS. The business WS security policy will be registered in the Kernel when the business WS wishes to use the security services provided by that Kernel. Hence, the Kernel 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'd like a simple message authentication based on X.509v3" certificates). There are some Intercommunication protocol between Kernel and business WS to coordinate the interactions of the different security services. On the other hand, the basic interactions are [1-12-15]:

• Registration/cancellation of the business WS in Kernel. A business WS must register itself in a Kernel including the definition of its Business Service Security Policy. This way the Kernel 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 Kernel or it could be forwarded by the business service to a Kernel so that the security service may be effectively applied.

At the end of this stage we should make the Security Architecture Specification document that shows us how the security scenarios display, through the architecture components interactions (Kernel 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 Kernel must perform one or more security requirements.

Outputs:

■ Appropriate WS-based security architectural patterns;
■ WS security policies associated with every Security Architectural Pattern;
■ Security Architecture Specification document;

C. Security Standards Identification

Inputs:

■ Security Architecture Specification document;
■ WS security policies identified in the stage Architectural Security Guaranty;

The main goal of SOA security standards is to provide a basis for interoperability among multiple products used in heterogeneous customer environments. Standards-based implementation strategies accelerate development, simplify integration, and reduce administrative costs over time. Most SOA industry standards are defined in XML frameworks. The last few years have seen the emergence of a plethora of XML-based specifications addressing various aspects of SOA security. Most of these specifications are part of the socalled WS-* (Web Services specifications) stack [13-15].

The following table shows which security requirements are satisfied by the various specifications and standards and how we can define Relationship between identified Web Service Security Requirements and Standards:

TABLE I
SECURE WS SPECIFICATIONS AND STANDARDS ADDRESSING

Dimension	Requirement	Specifications
Messaging	Confidentiality and Integrity	WS-Security
		SSL/TLS
	Authentication	WS-Security Tokens
		SSL/TLS X.509 Certificates
Resource	Authorization	XACML
		XrML
		RBAC, ABAC
	Privacy	EPAL
		XACML
	Accountability	None
Negotiation	Registries	UDDI
		ebXML
	Semantic Discovery	SWSA
		OWL-S
	Business Contracts	ebXML
Trust	Establishment	WS-Trust
		XKMS
		X.509
	Trust Proxying	SAML
		WS-Trust
	Federation	WS-Federation
		Liberty IDFF
		Shibboleth
Security Properties	Policy	WS-Policy
	Security Policy	WS-SecurityPolicy
	Availability	WS-ReliableMessaging
		WS-Reliability

Outputs:
- Appropriate WS security standards;
- Security Policies Specification;

III. RELATED 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 suggested method in this paper, 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.

IV. CONCLUSION

This paper has presented the security method, which allows us to provide a WS based service oriented architecture with security through a systematic stages. Some of the main aspects to be developed are the following: to complete the repository defined in the Initial Security Analysis 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 secure web service based SOA.

ACKNOWLEDGMENT

The author would like to acknowledge Prof. Dr Mir Ali Seyyedi, Department of Computer Software Engineering, Islamic Azad University, and anonymous reviewers for their invaluable suggestions and encouragement.

REFERENCES

[1] Rasmussen R E, Eggen A and Haakseth, "An architecture for experimenting with secure and dynamic Web Services", Proceedings of the 2006 Command and Control Research and Technology Symposium, San Diego, USA, 2006.
[2] ENDREI, M., ANG, J., ARSANJANI, A., CHUA, S., COMTE, P., KROGDAHL, P., LUO, M. and NEWLING, " Patterns: Services oriented architectures and web services", 2004.
[3] BASS, L., CLEMENTS, P. and KAZMAN, "Software architecture in practice", A 2003.
[4] Emig, C., Weisser, J., Abeck, S. "Development of SOA-Based Software Systems – an Evolutionary Programming Approach", In: IEEE Conference on Internet and Web Applications and Services ICIW'06, Guadeloupe / French Caribbean, February 2006.
[5] Newcomer, E., Lomow, G, "Understanding SOA with Web Services", Addison Wesley Professional, Reading , December 2004.
[6] Nadalin, A., Kaler, C., Monzillo, R., Hallam-Baker, P. (eds.), "Web Services Security (WSSecurity)", Version 1.1, February 2006.
[7] M. Tatsubori, T. Imamura, and Y. Nakamura, "Best Practice Patterns and Tool Support for Configuring Secure Web Services Messaging", IEEE International Conference on Web Services (ICWS), 2004.
[8] D. K. Barry, "Web Services and Service-Oriented Architectures", The Savvy Managers Guide, Morgan Kaufman Publishers, San Francisco, USA, 2003.
[9] M. Tatsubori, T. Imamura, and Y. Nakamura, "Best Practice Patterns and Tool Support for Configuring Secure Web Services Messaging", IEEE International Conference on Web Services (ICWS), 2004.
[10] PAPAZOGLOU, M. P. and GEORGAKOPOULO, "Service-oriented computing", Communications of the ACM, December 2004, 46 (10): 25-28.
[11] ALBERTS, C. J., BEHRENS, S. G., PETHIA, R. D. and WILSON, "Operationally critical threat, asset, and vulnerability evaluation (OCTAVE) framework", Version 1.0., Carnegie Mellon, Software Engineering Institute, 2005.
[12] SMITH, D. "Common concepts underlying safety, security, and survivability engineering", Carnegie Mellon, Software Engineering Institute, 2003.
[13] OMG, "UML profile for QoS and fault tolerance", see http://www.omg.org/docs/ptc/04-09-01.pdf, 2004.
[14] BASS, L., BACHMANN, F., ELLISON, R. J., MOORE, A. P. and KLEIN, "Security and survivability reasoning frameworks and architectural design tactics", Carnegie Mellon, Software Engineering Institute, 2004.
[15] KLEIN, M. and KAZMAN, "Attribute-based architectural styles", Carnegie Mellon, Software Engineering Institute, 2004.

Nafise Fareghzadeh (F'1983) was born in Tehran in Iran, on July 9, 1983. She graduated from the South Tehran Branch Islamic Azad University with the degree of M.Sc of software computer engineering. She is a is the associate professor at Computer Department of Islamic Azad University of Zanjan and Tehran, Payame Noor University of Tehran, , Raja Uniersity of Quazvin. She has been a computer software engineer with several software projects in Iran Railway and other Organization since 2003. She is author of two books. Furthermore, he has published several technical papers in conferences and journals.