Secure Document Circulation Using Web Services Technologies

Shane Bracher*

Bond University, Gold Coast QLD 4229, Australia
Siemens AG (Corporate Technology), Otto-Hahn-Ring 6, 81739 Munich, Germany
sbracher@student.bond.edu.au

Abstract. This paper discusses two issues. The first is the need for model driven security for Service-Oriented Computing environments to address the lack of support for specifying the security requirements of Web Services during the earlier phases of the development process. The second issue revolves around supporting secure document circulation in inter-domain, decentralized environments, and specifically, how to use model driven security and Web Services technologies to realize the design and implementation of this respectively. It is aimed that addressing both of these issues will contribute toward providing security for inter-organizational workflows which spread across multiple domains.

1 Introduction

Service-Oriented Computing has experienced considerable momentum in recent years as a new approach towards distributed computing. In this paradigm, distributed interacting entities are implemented as *services*. These are simply autonomous, platform-independent software components with the purpose of providing interoperability and collaboration within heterogeneous environments. Designing and implementing such an infrastructure can be achieved using Web Services. A strong advantage of Web Services is that it provides a standards-based, loose-coupling approach for combining multiple services (offered by different organizations) into a single, more sophisticated, value-added, composed service. The formation of these composed services is realized in terms of workflows (whereby control-flow and data-flow is specified).

A serious concern of Web Services is security. Increasing this concern is the fact that workflows can spread across domains. This complicates the issue as the security context now shifts from a single, centrally administrated domain to an inter-domain, decentralized environment. Although there is much work in existence for Web Services Security - such as WS-Security and the Web Services Security Stack [8], SAML [7], and XACML [10] - this is based on the implementational level. As for the design phase, and specifically "early Service-Oriented Computing design", support for security is lacking.

* Currently completing internship at Siemens Corporate Technology.

A promising approach is model driven security. The objective here is recognition of non-functional requirements (such as security) early in the development process. Considering such requirements at a later stage makes it (1) harder to integrate security measures into the product, and (2) increases the potential for greater security vulnerabilities arising in the product. By taking a model driven approach, this has the benefit of representing security requirements in an implementational-independent fashion, as well as providing the possibility for use with model checking tools for performing verification. Therefore, extending this model driven security approach to Web Services, and Service-Oriented Computing in general, would certainly be a step closer to addressing the security concerns of Web Services.

To demonstrate how Web Services Security can leverage the benefits of model driven security, the case of secure document circulation in inter-domain, decentralized environments is put forward. What is interesting about this case is that both it and Service-Oriented Computing share the same context - that is, distributed, heterogeneous, decentralized environments. As a result, the security issues here are comparable. Given the similarities, the question is can we achieve the objective of secure document circulation in inter-domain, decentralized environments by using Web Services technologies? More precisely, the goal is to determine if the work done towards Web Services Security can be applied to the securing of inter-organizational document flow (with the added intention of applying model driven security during the design phase).

Workflow is also an important component of secure document circulation, and this too is an issue of Service-Oriented Computing (in particular, service composition). Although this issue will be investigated in the thesis, the current focus is on security. Similarly, it is anticipated that the work of Web Services can also be applied in this regard (for example, the technologies for Web Services Orchestration).

The reminder of this paper is organized as follows: Section 2 discusses related work in model driven security; Section 3 provides further details on the research problem; and finally, Section 4 discusses future work for the thesis and concludes the paper.

2 Related Work

The issue of model driven security in the context of access control infrastructures is currently being explored by Basin et al. [2]. Their objective is to use modeling techniques to specify the security requirements of access control systems and then to automatically generate such systems from the models. The modeling language proposed is SecureUML [5] - a Unified Modeling Language (UML) based language for modeling Role Based Access Control (RBAC) policies. Much work has been done on using SecureUML for static UML diagrams (specifically, class diagrams), but as for supporting dynamic diagrams (e.g. use case diagrams, activity diagrams and sequence diagrams), this appears to be absent. For applying access control restrictions to workflows (e.g. for making flow

path decisions), support for access control policies in dynamic UML diagrams is necessary in order to specify this. Furthermore, this work limits itself to only access control and not other security properties.

The SECTINO project [4] takes an alternative approach by proposing a model driven security architecture. This architecture consists of a set of "model views" which are aimed towards representing workflows and associated security properties, but on an abstract level. The SECTINO project also focuses on UML and provides support for specifying confidentiality, integrity and non repudiation properties. Furthermore, automated "model to code transformation" is a key goal of this project. Although a broader spectrum of security properties are supported, current work focuses on class diagrams and activity diagrams. The fact that support for use case diagrams is lacking is significant given that the previously mentioned diagrams are derived from use case diagrams.

A promising development towards model driven security is the work of Jrjens et al. on UMLsec [6]. This is an extension to UML which allows for security requirements to be integrated into UML models for security-critical systems. The goals of UMLsec are to consider security requirements from the early design phases and to enable UML models to be evaluated for vulnerabilities. Hence, automated verification is a key aspect of this work. To model security requirements, UMLsec uses a set of pre-defined stereotypes and tags to label components in the model. Although consideration is provided for use case specifications in this work, further support for use case diagrams would be advantageous - especially for signifying the "scope" of the requirements.

3 Problem Description

For model driven security, this research aims to investigate existing works relating to extending the UML specifications with security specific elements so that security goals and security measures can be represented within UML models. The initial desire is to look at how the UML use case diagram can be augmented with security-based model elements. This will involve studying current proposals to determine what is missing in terms of security for Service-Oriented Computing environments, and then where necessary, proposing new security extensions to address this issue.

The types of security concerns that are of particular interest for modeling include message-level security, access control restrictions and outsourced security measures. As other UML diagrams are often derived from the information contained in use case diagrams, the objective is to formulate security constructs so that they can be easily propagated to subsequent UML diagrams (such as class diagrams, activity diagrams and sequence diagrams). Hence, this allows for reusability and consistency of the security constructs in the UML model.

The current vision for describing security requirements at the level of use case diagrams includes the following:

– Specifying security at the level of systems (all use cases must follow the requirements stated at this level).

- Specifying security at the level of use cases (defines the requirements which are specific to the given use case within a subsystem).
- Specifying security issues which are handled by trusted third parties. For example, authentication of users may be outsourced to identity providers (in order to support single sign-on).
- Specifying security requirements on messages related to externally handled security issues. For example, for an authentication token to be accepted, it must be digitally signed by a recognized identity provider. Essentially, this allows us to state the security requirements of the systems interface.

It is important to emphasize that this vision considers a Service-Oriented Computing environment. Therefore, this serves towards addressing the concern of a lack of security support for the design of Web Services during the development cycle. For added support, a further step of this work could look at verification techniques for verifying that the modeled security measures meet the security goals of the service. This would effectively provide security assurances that the service is resistant to certain attacks. Experimenting with security verification tools such as the AVISPA toolset [1] and its associated specification language, HLPSL [3], is one possibility for conducting this.

To apply these ideas of model driven security for Service-Oriented Computing, the case of supporting secure document circulation in inter-domain, decentralized environments will be explored. The objective here is to investigate a new approach for providing enhanced document protection designed for distributed, heterogeneous, decentralized environments. In terms of security, this is a non-trivial issue as multiple domains are involved. Therefore, we cannot rely on traditional security mechanisms which assume the existence of a central administration authority. As an alternative approach, security requirements will be enforced by mechanisms embedded within the document itself. The vision is the realization of a "smart" or "intelligent" document - one with richer capabilities regarding security and workflow, all within an inter-domain context.

In relation to implementing access control, this has suddenly become complicated due to the absence of a single enforcement point. In fact, the only way to provide access control in this situation is through encryption. This now results in a shift in policy specifications from "what actions is a user allowed to perform" to "who has access to the decryption key". This increased reliance on encryption introduces the need for finer granularity for confidentiality and data integrity of the document's content. Furthermore, it also introduces the additional challenge of needing to provide key protection strategies.

The current proposed framework for the smart document is displayed in Figure 1. This is simply a high-level, implementation-independent view of the smart document design. At the core of this framework are the document and document metadata components. The document component represents an ordinary document containing content. The document metadata component provides the document with, amongst other things, enhanced security capabilities. Workflow and state awareness are also planned to be provided by this component. Associated with the document metadata component are four sub-components. In

order, these sub-components provide flow awareness, state awareness, security policies and additional attributes required for fulfilling the security and workflow functionalities. The description components attached to the document and document metadata components provide structure and content control.

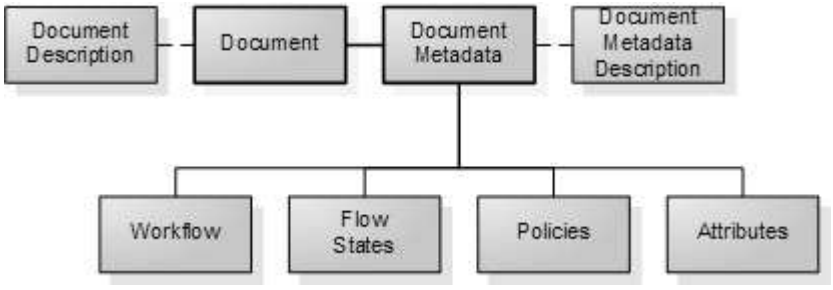

Fig. 1. High-level framework view of the smart document design.

4 Future Work and Conclusion

On the completion of the discussed work on model driven security and the design for the smart document framework, the next step is to see how Web Services technologies can be used for implementing the smart document design. For implementing the security functionalities, current technologies of interest include XML-Encryption [11], XML-Signature [12], SAML [7], XACML [10] and WS-SecurityPolicy [9]. However, this is only a preliminary list and no doubt additional technologies may be needed. Using the above mentioned technologies, the aim is to implement finer granularity document confidentiality and data integrity, decentralized access control enforcement, and security policies. After an implementation prototype has been completed, the final step will be to conduct a case study to show a real world application of secure document circulation using Web Services technologies.

In conclusion, the objectives of this research are two-fold. Firstly, it aims to address the need for security requirement specification support during the earlier stages of the development process. In particular, the desire is to focus on Service-Oriented Computing environments for this. Secondly, this research aims to apply the first objective into a more practical scenario - namely, using Web Services technologies to implement secure document circulation in inter-domain, decentralized environments. Given that the context and the issues affecting this scenario and Service-Oriented Computing are quite similar, it seems worthwhile to envision the design and implementation of this scenario in terms of Web Services technologies.

Acknowledgements

I would like to thank my supervisors Dr Paddy Krishnan, Dr Jorge Cuellar and Dr Zheng da Wu for their assistance and guidance towards my PhD candidature to date. Also, I thank my workgroup at Siemens Corporate Technology for their support and feedback on my research - particularly to our visiting colleague, Dr Shmuel Tyszberowicz of Tel Aviv University, for the many helpful discussions on model driven security.

References

1. Automated Validation of Internet Security Protocols and Applications. `http://www.avispa-project.org/`. September 6 2005.
2. D. Basin, J. Doser, and T. Lodderstedt. Model Driven Security: from UML Models to Access Control Infrastructures. *ACM Transactions on Software Engineering and Methodology*, 2005. to appear.
3. Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, J. Mantovani, S. Mödersheim, and L. Vigneron. A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols. In *Workshop on Specification and Automated Processing of Security Requirements (SAPS 2004)*, 2004.
4. M. Hafner, R. Breu, and M. Breu. A Security Architecture for Inter-Organizational Workflows: Putting Security Standards for Web Services Together. In *ICEIS 2005, Proceedings of the Seventh International Conference on Enterprise Information Systems, Miami, USA, May 25-28, 2005*, pages 128–135, 2005.
5. T. Lodderstedt, D. A. Basin, and J. Doser. SecureUML: A UML-Based Modeling Language for Model-Driven Security. In *UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language*, pages 426–441, London, UK, 2002. Springer-Verlag.
6. G. Popp, J. Jürjens, G. Wimmel, and R. Breu. Security-Critical System Development with Extended Use Cases. In *APSEC '03: Proceedings of the Tenth Asia-Pacific Software Engineering Conference Software Engineering Conference*, page 478, Washington, DC, USA, 2003. IEEE Computer Society.
7. OASIS Security Assertion Markup Language (SAML) Version 2.0. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security`. March 15 2005.
8. Security in a Web Services World: A Proposed Architecture and Roadmap. `ftp://www6.software.ibm.com/software/developer/library/ws-secmap.pdf`. April 2002 Version 1.0.
9. Web Services Security Policy Language (WS-SecurityPolicy). `http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf`. July 2005 Version 1.1.
10. OASIS eXtensible Access Control Markup Language (XACML) Version 2.0. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml`. February 1 2005.
11. XML Encryption Syntax and Processing. `http://www.w3.org/TR/xmlenc-core/`. December 2002.
12. XML Signature Syntax and Processing. `http://www.w3.org/TR/xmldsig-core/`. February 2002.