

Message-level access control for service-oriented computing

Kees Leune and Bart Orriens

Tilburg University, Infolab
{kees,borriens}@uvt.nl

Abstract. Service-oriented computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing applications. In order to realize the vision of services being utilized as fundamental elements of business collaboration, security is a critical issue that must be addressed. As SOC adopts messages as the binding factor for loosely couple service interactions, messages must be protected in the context in which they are exchanged. In this paper, we adopt a layered security approach in which we argue that transport level security and message level security, such as provided by current web services standards and products, are not adequate for protecting services from unauthorized use, but that an additional layer in which the message context is protected is required. The message context layer is subsequently related to the EFSOC framework, which provides a event-driven framework for service-oriented computing.

1 Introduction

Recently there has been increasing focus on service-oriented computing (SOC), the new emerging paradigm for distributed computing and e-business processing, to deliver flexible and adaptable corporate business services by utilizing existing services across organizational boundaries. *Business collaboration* refers to a cooperation between multiple enterprizes working together to achieve a common business goal. In order to realize the vision of utilizing services as fundamental elements for developing applications [10] for business collaboration, security is a critical issue that must be addressed. Businesses will be averse to participating in cooperations that do not take place in a trusted environment, for example to avoid problems concerning denial of actions, unauthorized reading of information, and so on.

Therefore, for the successful adoption of SOC within the business collaboration domain, the paradigm must provide the means to make cooperation between enterprizes secure. At the moment, the most successful manifestation of SOC can be found in web services technology. A web service is a specific kind of service that can be unambiguously identified (generally by means of a URI) and whose service description and transport utilize open Internet standards, such as XML-based SOAP messages. Unfortunately, most work in the web service security arena concentrates on privacy and integrity of messages, and to a

lesser extent on authentication and authorization. In this paper we introduce the EFSOC framework, which is currently being developed at Tilburg University's Infolab. This framework provides an event-driven access control approach for authorization in the context of service oriented computing.

The remainder of this paper is structured as followed: in section 2 we discuss business collaboration and the role of security. Then, security within service-oriented computing and the current work in this area is analyzed in section 3, where we identify the gap existing with regard to authorization. Subsequently, we present the EFSOC framework in section 4 to address this void. Finally, we present conclusions in 5 and outline future work.

2 Security in Business Collaboration

Security in general can be viewed as being concerned with establishing the capacity "to be able to avoid being harmed by any risk, danger or threat" [4]. Interpreted in the context of business collaboration security deals with providing assurance to enterprises that their cooperation is taking place in a secure manner. In other words, security is concerned with providing peace of mind for the businesses involved, where they can rely on the fact that their collaboration is safe from risks like impersonation, unauthorized use of resources, an etceteras. Now, in business collaboration security can be perceived at different levels, being *business*, *conceptual* and *logical* level, each of which represents a level of abstraction with its own content and meaning regarding security. In the remainder of this section we shall briefly introduce each level and discuss the role of security at this level. An overview is provided in Figure 1.

2.1 Business Level

At an abstract business level a business collaboration constitutes a cooperation between enterprises making use of each other's business services to exchange resources to further their business goals. At this level security deals with the analysis of threats that can jeopardize the successful completion of these resource exchanges. Threat analysis here is aimed at 1) identification of the threat, 2) measuring the magnitude of the potential loss if this threat is not dealt with, and 3) the probability that the loss will occur. For business collaboration commonly six types of security threat are identified:

1. *Masquerading*

A first threat is that of masquerading, which is the threat of an entity pretending to be another entity. This can involve one business collaboration participant assuming the identity of another participant. Alternatively, it can be the case that an outside party is trying to infiltrate the collaboration by stealing the identity of one of its participants.

2. *Unauthorized Access*

Masquerading is usually the means to an end for attackers to gain access to resources/services, information, etc. to which they are ordinarily not entitled.

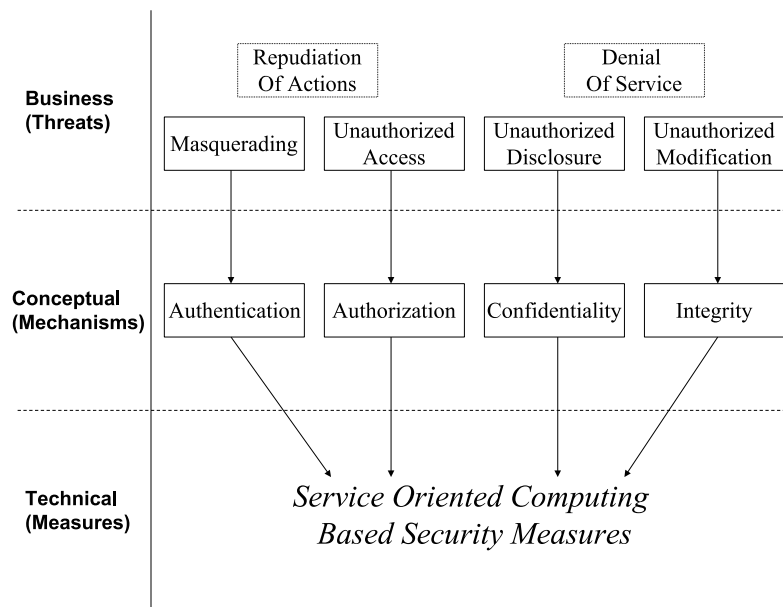


Fig. 1. Security in Business Collaboration

In this sense masquerading is related to the risk of *unauthorized access*, which pertains to the usage of resources/services by business collaboration participants or outside parties who are not allowed to do so.

3. *Unauthorized Disclosure*

A third threat in the business collaboration environment is concerned with the disclosure of information to unauthorized parties. Information exchanged among business collaboration participants is often of a sensitive nature, e.g. electronic patient records, insurance claims, payment roll information, etc. It is not difficult to see that the consequences of such private information falling into wrong hands could spell disaster.

4. *Unauthorized Modification*

Rather than dealing with the wrongful disclosure of information the fourth threat, unauthorized modification, focuses on the alteration of information as it is being exchanged between business collaboration participants. Think for example of some party making unauthorized changes to your product order, or more extreme modifying blueprints of the airplane you are constructing.

5. *Repudiation of Actions*

Repudiation of actions is the fifth threat, and expresses the danger against accountability where business collaboration participants can deny having performed certain actions. Typical situations where repudiation is an issue, is in the domain of bank transactions, bid offers in auctions, and so on.

6. *Denial of Service*

The sixth and final threat is denial of service, capturing the possibility that authorized participants are unable to access a business service due to unavailability. Another option is that requests for service usage are blocked or delayed, which can be viewed as an extreme case of unauthorized modification.

In this paper we focus on prevention of the first four security threats. The fifth and sixth threat are not taken into consideration, as the defensive arrangements required for their neutralization are partially outside the scope of the field of security (such as monitoring and logging for repudiation, and load balancing, service pooling, etceteras for denial of service).

2.2 Conceptual Level

Based on the threat analysis at the business level security measures are determined at the conceptual level to establish defenses capable of tackling the identified threats. Because at this perspective a business collaboration constitutes the sending and receiving of appropriate documents by enterprises to make use of their respective business services, operationalization of security threats entails specifying the security mechanisms that will be employed to provide protection for this document communication. In correspondence with the described threats in subsection 2.1 the following security mechanisms can be utilized:

– *Authentication* → *Masquerading*

Countering identity theft requires an unambiguous way in which the identity of a participant can be established. For this purpose an authentication mechanism is employed, which revolves around using a combination of something a business collaboration participant knows, has and possesses to check its identity claim. Authentication can be done directly by the participant interested in verification of the identity, or indirectly where this process is delegated to a trusted third party.

– *Authorization* → *Unauthorized Access*

In order to be able to verify access to business services, it must be clear who is allowed to what. This necessitates the presence of an authorization mechanism, which limits and controls access to information and resources by addressing the range of things a participant is allowed to do. The most common approaches for authorization are grounded on a discretionary, mandatory or role based scheme [7]. Authorization often requires authentication, however, this is not always the case. For example, entrance tickets give you the right to access something, but do not require any identification by themselves.

– *Confidentiality* → *Unauthorized Disclosure*

The security mechanism used to prevent unauthorized disclosure of resources is confidentiality, which helps avoid situations where unauthorized parties can observe information. Confidentiality is usually associated with some form of encryption mechanism, where a distinction is often made between cryptographic and hashing techniques. Cryptographic techniques provide means of converting information from its normal, comprehensible form into an incomprehensible format, rendering it unreadable without secret knowledge. Hashing techniques produce a document digest, which appears random to attackers and does not leak information about the content itself.

– *Integrity* → *Unauthorized Modification*

In order to ensure that information can not be modified during its exchange without this being notice, the mechanism of integrity is required. Integrity deals with ensuring that any alteration to a resource will be detectable. This mechanism is passive in the sense that modification can not be prevented. Rather its purpose of detection is focused on identifying modifications based upon which appropriate action can be taken. Integrity, like confidentiality, often utilizes some form of hashing technique. Because the digest produced in hashing is (in theory) unique for any document, any change (even a single letter) will result in a different digest. Checksum algorithms like cyclic redundancy checking are another option, providing a simple form of protection against document modifications.

Together these four security mechanisms provide the building blocks required to tackle the threats present in the business collaboration environment. Observe that these mechanisms are often used in conjunction to further strengthen security defenses, for example using confidentiality and integrity to protect authentication or authorization information, or utilizing authentication to first establish identity upon which an authorization mechanism is grounded.

2.3 Logical Level

The implementation of the security mechanisms selected at the conceptual level must subsequently be implemented at the logical level via security measures. This level is the domain of the service oriented computing paradigm. In this paradigm a business collaboration is viewed as a set of interacting technical services, where these interactions are message based and facilitate the communication of information among collaboration participants. In order to meet the business driven security demands at this level, the message based interactions, i.e. message exchanges, must be adequately protected. In the next two sections we explore service oriented computing in more detail, where we particularly focus on how the application of security measures in the context of message exchanges between services can be accommodated.

3 Security and Service-Oriented Computing

As explained in the previous section security requirements are driven by business needs. Furthermore, it was observed that from a service oriented computing perspective, these needs require that the message based interactions between services must be adequately protected (as service-oriented computing in general, and web services technologies in particular, are message centered). On a computer network these services interactions can often not be distinguished from regular web traffic, since the majority of web services communicate via SOAP messages. Because such messages are often transported using the HTTP protocol connecting to “normal” web server network ports, many service interactions bypass most corporate firewalls without intervention. For this reason each SOAP message is a potential security threat.

Figure 2 subdivides the security field in three closely related categories. Transport level security addresses the protection of the message transport channels. The transport layer in this paper roughly corresponds to the first five layers (up to the session layer) of the seven-layer OSI reference model. Provided that they are adequately deployed, technologies like SSL provide a basic level of protection. However, transport level security is not enough, as message exchanges are still vulnerable to attacks which focus on the transport layer end-points, such as man-in-the-middle attacks.

Message level security protects individual messages. Even when messages are intercepted as they are being transferred over a secured transport layer, the message content will not be understandable to anyone else than the intended recipient. However, even with message-level protection in place, an attacker can still intercept messages, and inject them back into the system at a later time. Such replay attacks can still lead to significant compromises. Protecting the content of message against compromise and against observation by improperly authorized subjects is typically achieved by deploying a number of approaches which supplement each other, such as digitally signing of messages and encryption of message content. Standards such as XML Signature [3] and XML Encryption [8] provide low-level support for signing XML messages. XML Signatures

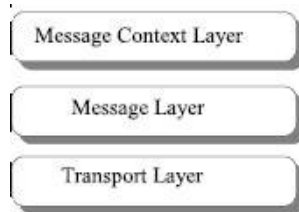


Fig. 2. A layered approach to security

provides integrity, message authentication and/or signer authentication services for data of any type. XML Encryption is a standard for a process for encrypting/decrypting digital content (including XML documents and portions thereof) and the XML syntax used to represent it. Both standards operate directly on XML documents, and are schema independent.

Web-services security standards, such as WS-Security [1], WS-SecurityPolicy, WS-SecureConversation, etc. provide more elaborate approaches. WS-Security, which relies on XML Signature and XML Encryption, proposes a standard set of SOAP extensions that can be used when building security Web Services to implement integrity and confidentiality, supporting multiple tokens, trust domains, signature formats, etc. WS-SecurityPolicy [6] provides the mapping of the WS-Security standards onto the infrastructure proposed by the WS-Policy standard [2]. WS-SecureConversation is built on top of the WS-Security and WS-Policy models to provide secure communication between services. WS-Security focuses on the message authentication model but not a security context, and thus is subject several forms of security attacks [5].

On the message context level, individual message are protected in the context in which they were sent. On this level, subject authentication information and session data is available. A message context is formed by a collection of metadata elements which describe each message in relation to other messages, or contain additional information about the environment in which it was sent. While single message authentication, such as provided by WS-Security, has its merits, a majority of business interactions can only be expressed as sequences of messages. Using a message context which is shared by all participants in such an interaction not only provides a powerful medium to express security constraints, but also provides a starting point for auditing purposes. The concept of a message context can be illustrated by the following example. In WS-Security, an explicit choice was made to support single message authentication. As such, it is not possible to related messages to one another to detect relationships between messages. To be able to specify policies such as “message b may only be sent in response to message a ”, contextual information is required.

All of the standards which have been discussed have in common that they focus heavily on privacy and integrity of single messages. The exception is WS Secure Conversation, which adds the concept of a security context which is shared by the participants of a business interaction. As such, the current standards do not address the business needs for security outlined in section 2, as they do not provide mechanisms required for authentication and authorization. In the following section we introduce the EFSOC model, and describe how its concepts can be used to implement authentication and authorization in service-oriented computing.

4 Access Control in EFSOC

The Event-driven Framework for Service Oriented Computing, in short EFSOC, comprises a layered framework to develop business processes with web services. The framework takes into account authorizations and roles that web-services may play during the course of a business process. EFSOC is a multi-level event-driven framework for service-oriented computing [9], consisting of an event layer, a security layer, and a business process layer, as depicted in figure 3.

In EFSOC, events are separated from access right capabilities that constrain the usage of events by subjects. In turn, the events themselves are isolated from the definition and enactment of associated services and processes. This facilitates not only specification and execution of events, processes and services, but also provides support for run-time decision making processes revolving around new or adapted security policies.

In accordance with the previous EFSOC constitutes of three layers (see also Figure 3): 1) the *event* layer, which provides brokered event-driven message exchange functionality to services; 2) the *security* layer, which adds security capabilities, with the emphasis on authorization and access control; and 3) the *business process* layer which provides the link between business processes requiring certain functionality, and the services that provide it.

Events play a pivotal role in the EFSOC framework as they serve as the elements providing the loose couplings between services. Events can be perceived as occurrences that happen over time, independently from each other [7]. An event thus constitutes an “occurrence” that causes, or is caused by, an ‘impact’ on a business process.

EFSOC events are represented by messages which consist of three parts. The message body represents the circumstances in which the event occurred and may contain any data structure. The message headers represent the context in which the message was sent, and contain elements such as timestamp, event body type, event generator, the relationship of the event to other events, etc. Finally, the event envelope may contain routing information which can be used to determine to which subjects an event will be delivered.

Separating the message body from the message headers provides an infrastructure which can be used to encrypt and sign the message body, independently

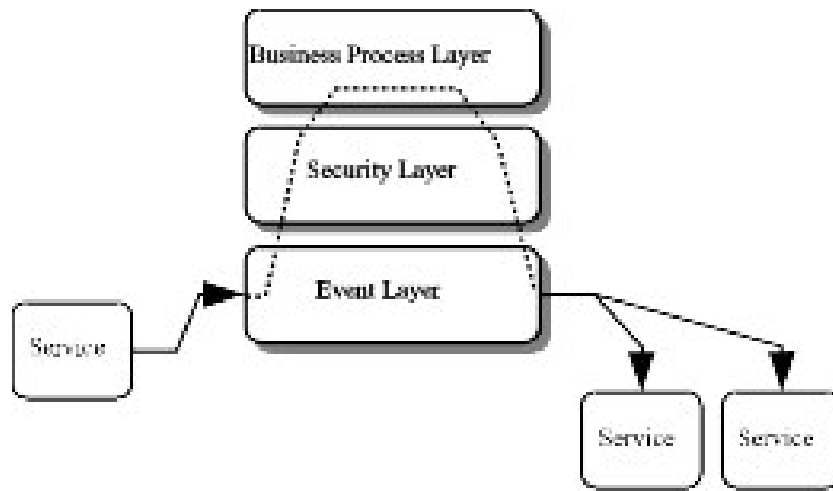


Fig. 3. EFSOC Framework

of the headers and the routing information. As such, the message content can be protected against unauthorized disclosure or manipulation.

EFSOC's security layer is positioned between the event-layer, which implements the event-driven messaging layer on top of the technical transport layer, and the business process layer, which maps events to services and workflows. As such, any service interaction needs to pass through the security layer.

The security layer provides constructs for specifying *access control constraints*. Transport level security and message level security (see figure 2) is assumed to be provided by existing technologies. Extending Sandhu (1996) [11], we adopt a role-based perspective to access control which we augment with temporary permission delegations. Access control policies can be specified on a static level, by which we mean on the level of role assignments (and, by transitivity, permission assignments), and on a dynamic level, by which we mean that it is possible to specify more finely grained runtime constraints.

The EFSOC framework is modeled around the idea that permissions are related to roles. Whether that relationship is one that is the result of a direct assignment of permission to roles, as is the case in most RBAC approaches, or if it is the result of a computational action which calculates the relationships, it will always be possible, at a given point in time, to determine the permissions associated with a role, and by extension, the permissions associated with a subject.

Subjects are assigned to roles and roles are assigned to permissions. Therefore, it is possible at any time to consider a role as a collection of subjects, as well as a collection of permissions. EFSOC adopts the viewpoint that explicitly formed relationships (i.e. assigned by some authority) between subjects and roles, and between roles and permissions, and between roles and responsibilities, are (semi-)permanent. In other words, when a subject is assigned to a role, it is intended that this is the case for a longer period of time.

However, semi-permanent relationships are an inhibiting factor for service-oriented computing, which strives in highly dynamic environments. To counter this limitation, subjects can delegate permissions and/or responsibilities on a temporary basis, provided that they have been explicitly assigned a role which is associated with those permissions and/or responsibilities.

The temporal nature of role delegations is constrained by any combination of:

- *A time interval*: The time interval constraint restricts delegations to take place, and to remain valid, in a certain period of time. Consider permission p_1 and a temporal constraint which states that delegations of the permissions can only take place, or remain valid, between January 15th 2005 14:00 and 15:00. This would imply that permission p_1 cannot be delegated before January 15th 14:00, or after January 15th 15:00. Additionally, any already delegated permissions will lose their validity after that time.
- *The maximal width of delegation*: The width of the delegation constraints the number of times that a subject can delegate the same permission. For example, it is conceivable that a delegation constraint is in place which

states that permission p_1 may only be delegated twice. For example, when the situation occurs that subject s_1 delegates to subject s_2 and subject s_3 delegated his permission to subject s_4 , the width constraint is met and no further delegation is possible.

- *The depth of delegation*: Likewise, the depth of delegation refers to the number of times that a specific permission can be re-delegated. For example, assume a situation where a delegation constraint exists which limits the depth of delegation of permission p_1 to two, and subject s_1 delegates to s_2 . Subject s_2 can now only delegate the permission once more before the delegation depth constraint is met.

EFSOC strongly prefers self-acted delegations. While, we acknowledge the fact that the initial assignment of roles to subject and of subject to roles is an activity which may be centrally coordinated, the need for decentralized (subject-initiated) delegation of assigned permissions is a critical success factor for the ability to rapidly respond to changing conditions.

In the context of the EFSOC framework, the rules for revocation follow the concept of active security. By this is meant that a permission must be revoked as soon as the basis for that permission is no longer valid. In other words, when a role is unassigned to a particular subject, all permissions that a subject received as a result of the role assignment must be revoked too, including any possibly delegated permission. This has a major implication on transactional aspects, since such a revocation can take place in the middle of a running operation. Although we acknowledge that this is a valid issue, transactional aspects are outside the scope of the EFSOC research.

5 Conclusions and Future Work

Current standards for security in service oriented computing have in common that they focus heavily on privacy and integrity of single messages (the exception being WS Secure Conversation, which adds the concept of a shared security context). These mechanisms are suitable for preventing unauthorized disclosure and unauthorized modification of messages, however, they do not offer support for authentication and authorization of service requesters and providers. Without support for prevention of masquerading and unauthorized access service oriented business collaboration can not take place in a trusted environment; something which is a key requirement of enterprises when considering cooperation with other organizations.

In this paper we have presented a multi-level event-driven framework for service-oriented computing called EFSOC, a framework that utilizes an event, security, and business process layer to establish security capabilities for authentication and authorization on top of the message-level security provided by existing standards, such as WS-Security. The event layer provides brokered event-driven message exchange functionality to services. The security layer adds security capabilities, with the emphasis on authorization and access control. Finally, the

business process layer provides the link between business collaborations requiring certain functionality to prevent particular security threats, and the services that provide it.

The main contribution of the work presented herein lies in the fact that the EFSOC framework fills the gap in the current work on security in service-oriented-computing with regard to masquerading and unauthorized access. Moreover, as EFSOC is standard agnostic, it may be used in combination with existing standard languages, such as WSDL and BPEL. Future work will be aimed at exploring the dynamic nature of event based business collaborations further by carefully studying dynamic events. In addition, we wish to examine modifications and maintenance to existing infrastructure and operational services in the EFSOC framework, allowing for pro-active change management support.

References

1. Atkinson, B., G. Della-Libera, S. Hada, M. Hondo, P. Hallam-Baker, J. Klein, B. LaMacchia, P. Leach, J. Manfredelli, H. Maruyama, A. Nadalin, N. Nagaratnam, H. Prafullchandra, J. Shewchuk, D. Simon, Web Services Security (WS-Security), <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>, 2002
2. S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, D. Langworthy, A. Malhotra, A. Nadalin, N. Nagaratnam, M. Nottingham, H. Prafullchandra, C. von Riegen, J. Schlimmer, C. Sharp, J. Shewchuk, Web Services Policy Framework (WS-Policy), <http://www-106.ibm.com/developerworks/library/specification/ws-polfram/>, September 2004
3. M. Bartel, J. Boyer, B. Fox, Brian LaMacchia, E. Simon, XML-Signature Syntax and Processing <http://www.w3.org/TR/xmlsig-core/>, 2002
4. Cambridge Learner's Dictionary, <http://dictionary.cambridge.org>
5. G. Della-Libera, B. Dixon, P. Garg, S. Hada, P. Hallam-Baker, M. Hondo, H. Maruyama, N. Nagaratnam, A. Nash, R. Philpott, H. Prafullchandra, J. Shewchuk, D. Simon, E. Waingold, R. Zolfonoon, Web Services Secure Conversation Language (WS-SecureConversation), <http://www-106.ibm.com/developerworks/library/specification/ws-secon/>, 2002
6. G. Della-Libera, P. Hallam-Baker, M. Hondo, T. Janczuk, C. Kaler, H. Maruyama, N. Nagaratnam, A. Nash, R. Philpott, H. Prafullchandra, J. Shewchuk, E. Waingold, R. Zolfonoon, Web Services Security Policy (WS-SecurityPolicy), <http://www-106.ibm.com/developers/library/ws-secpol/>, 2002
7. W. van den Heuvel, K. Leune, M. Papazoglou, EFSOC: A Layered Framework for Developing Secure Interactions between Web-Services, *Kluwer Academic Publishers, Vol. 13, No. 12, pp. 1-38, 2005*
8. T. Imamura, B. Dillaway, E. Simon, XML Encryption Syntax and Processing, <http://www.w3.org/TR/xmlenc-core/>, 2002
9. K. Leune, M. Papazoglou, W. van den Heuvel, Specification and Querying Security Constraints in the EFSOC Framework, *Proceedings of the 2d International Conference on Service Oriented Computing, New York City, USA, 2004*
10. M. Papazoglou, G. Georgakopoulos, Introduction to the Special Issue about Service-Oriented Computing, *Communications of the ACM, Vol. 46, No. 10, pp. 24-29*
11. R. Sandhu, E. Coyne, H. Feinstein, C. Youman, Role-Based Access Control Models, *IEEE Computer, 1992*