

Expressing Authorization in Semantic Web Services

Richard Scott Patterson, John A. Miller

Abstract - Currently many Web services use authentication to make authorization decisions on an operation by operation basis. Semantic Web services need the ability to describe authorization for the purpose of Web service discovery. In this paper we propose a framework for describing authorization using Semantic annotations in WS-Policy. These annotations are used in the post-‘Semantic Discovery Phase’ to determine if a requester has permissions to invoke a published service.

I. INTRODUCTION

Web Services continue to evolve and impact the ways in which businesses interact with the world. Semantic Web services, where agents or discovery engines select appropriate services to invoke on behalf of the user, will have a substantial impact on businesses and individuals. Authorization is a challenge to the Semantic Web services environment.

Authorization is determining who has permission to use which resources. In an organization where the users and resources are known to those creating the authorization scheme, this can be problematic but is feasible. However, Web services are exposed to the world over the internet and it would not be good business practice to allow anonymous access to resources. Currently many organizations ‘spell out’ in human readable form which requestors will have access to its services. For Semantic Discovery of Web services there needs to be a framework which will indicate authorization for a service.

The work presented in this paper focuses on Semantic description of authorization for Web services. In particular, Semantic annotations, which are extensible elements and ontological concepts, are added to WS-Policy in order to describe authorization to invoke a Web service. This aids in the Semantic discovery of Web services by allowing potential requestors, or users, to perform Semantic matching in an effort to determine if they have authorization for a service.

A. Web Services Security Background

Web services security encompasses securing the system running a Web service, securing the service itself, securing the

network on which the service system is running, securing the communications between the requestor and service provider, authenticating both parties, and authorization of requestors[10]. System and network security are not new to the security space and application security remains an issue for the developers. However, securing communication and authentication in a distributed and dynamic environment requires a new approach. Much of this has been accomplished through the WS-Security [9] and WS-Policy [11] specifications. WS-Security is specific to SOAP messaging, providing a standard for integrity, confidentiality, and authentication of SOAP messages. WS-Policy provides a set of constructs that can be used to describe functional and non-functional capabilities and requirements. It is for this reason we chose to add the annotations to WS-Policy.

SAML, Security Assertion Markup Language, is an OASIS specification in which assertions about a subject are issued by authorities such as certificate authorities, and attribute authorities. Digital Certificates are issued by certificate authorities and contain information like a name, an expiration date, and a public key.

The current authorization scheme for Web Services is depicted in Figure 1. When a Requestor discovers a Web Service, either by UDDI or Web site, a human readable description is given for authentication and authorization, usually this consists of login requirements or account creation instructions. When the requestor attempts to invoke the service, the Provider decides whether or not to grant permission based on the information passed during authentication.

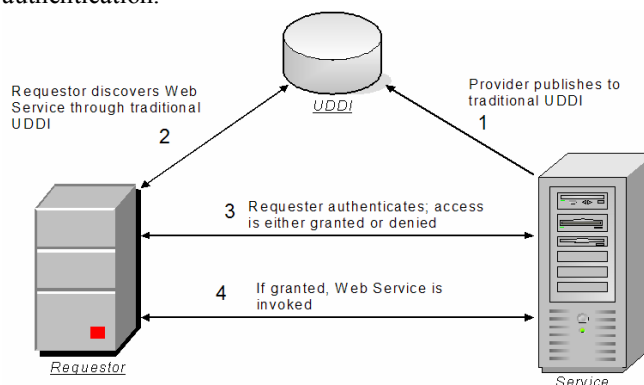


Figure 1 - Current Authorization Scheme

Manuscript received January 25, 2005.

R. S. Patterson is with the LSDIS Lab of the Computer Science Department at The University of Georgia, Athens, GA 30602 USA (rsp@cs.uga.edu).

J. A. Miller is a faculty member of the LSDIS Lab of the Computer Science Department at The University of Georgia, Athens, GA 30602 USA (jam@cs.uga.edu).

II. SEMANTIC AUTHORIZATION

Authentication of requestors in the Web services space has been addressed. Our Focus is on the Semantic representation of authorization and access control information for Web service discovery. We will assume that when a requestor has been authenticated, the service provider believes they are who they say they are. However, this is not granular enough to grant permission to many of the resources available through Web services. Furthermore each provider of a Web service may have different criteria for granting access and this needs to be expressed in order to discover those services a requester has permissions to invoke.

Our approach is to add Semantic annotations to WS-Policy which describe the authorization criteria. These annotations are used by the requestor in a post-‘Semantic Discovery Phase’ to perform Authorization Verification. The constraint analysis is based on a Client Authorization Cache (CAC), Web Services Policy – Semantics (WSP-S), Rule Based Access Control (RBAC) ontology, and any relevant Domain ontologies. We will expand upon each of the above points to our approach starting with annotations.

A. Annotation - RBAC Extension Elements

The annotations are ontological concepts and express what type of requestor has authorization to use the associated Web service; all else is an implicit deny. (*It is important to note that this is not an enforcement point of authorization, rather an aid to Semantic discovery.*) For example, the ontological type could be a requesters title, an organization to which they belong, or a group in which they are a member.

The annotations have extensibility elements similar to the extensibility elements provided in WSDL-S [12], like precondition and effect. WSDL-S is a W3C specification submission which supports semantic representation in WSDL through annotations. WSP-S will provide extensibility elements for semantic representation of authorization. The extensibility elements are derived from the RBAC standard [13] and eXtensible Access Control Markup Language (XACML) representation of RBAC. [14] XACML is an OASIS XML schema for representing authorization and entitlement policies

We chose the RBAC standard because it is widely accepted, easily understood, and succinctly expresses authorization permissions. However, the enforcement point for authorization does not need to be implemented using RBAC and can be implemented in anyway suitable to the provider of a Semantic Web service.

Here we will cover the extensibility elements, their descriptions, and give examples. Our first extension element is *subject*. *subject* is equivalent to a *user* in RBAC. According to the National Institute of Standards and Technology a *user* is defined as a human, machine, network, or intelligent autonomous agent. In our context, this can include an entire organization. An individual such as Fred Jones can be a *subject*, as can General Motors.

The extension element *role* is a function within the context of an organization; some associated semantics regarding the authority and responsibility are conferred on the user assigned to the role. A role could be general, for example employee, or more specific as in field engineer.

```
<wsp:Policy
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsse="http://docs.oasis-open.org/wssecurity-secext-1.0.xsd"
  xmlns:qos = "http://www.w3c.or.kr/kr-office/TR/2003/ws-qos"
  xmlns:wspsem="http://www.ibm.com/xmlns/WebServices/WSPolicySemantics"
  xmlns:Ontology0="http://lstdis.cs.uga.edu/projects/meteor-s/wSDL-s/ontologies/rbac.owl"
  xmlns:Ontology1="http://lstdis.cs.uga.edu/projects/meteor-s/wSDL-s/ontologies/pips.owl"
>
  <wsp:ExactlyOne>
    <wsp:All>
      <wspsem:subject name="Organization" wspsem:modelReference="Ontology1#Organization">
        <wspsem:role name="inventory_manager" wspsem:modelReference="Ontology0#inventory_manager">
          <wspsem:subjectCategory name="Partner" wspsem:modelReference="Ontology1#Partner"/>
          <qos:cost unit="dollars"> 10 </qos:cost>
          <wsse:SecurityToken>
            <wsse:TokenType>wsse:Kerberosv5TGT</wsse:TokenType>
          </wsse:SecurityToken>
        </wsp:All>
      <wsp:All>
        <wspsem:subject name="PreferredClient" wspsem:modelReference="Ontology1#PreferredClient">
          <wspsem:subjectCategory name="Partner" wspsem:modelReference="Ontology1#Partner"/>
          <qos:cost unit="dollars"> 8 </qos:cost>
          <wsse:SecurityToken>
            <wsse:TokenType>wsse:X509v3</wsse:TokenType>
          </wsse:SecurityToken>
        </wsp:All>
      </wsp:All>
    </wsp:ExactlyOne>
  </wsp:Policy>
```

Figure 2 - Annotation Example

The extension element *subjectCategory* describes the type of subject. For example, partner describes that the subject is in some kind of partnership agreement with the provider of the Web service.

Lastly, *modelReference* is used to handle one-to-one mapping of schema elements to an ontological concept. For example, this can be applicable when a Web service provider wants to demonstrate that authorization will be constrained to certain inputs for an operation. This might be done using an ontological concept like *manufacturers_Part_Number*.

This annotation scheme provides the granularity needed for Web services. This is because a WS-Policy file may be attached to a message, a service binding, an operation, or a parameter such as an input. The annotations are used to describe an explicit ‘grant’; while lack of the criteria or conditions is an implicit ‘deny’.

B. Authorization Verification

Once Semantic Discovery has returned a set of candidate services, the requestor, or an agent acting on behalf of the requestor, can perform constraint analysis to determine which of the candidate services it most likely has authorization to invoke, Authorization Verification. Constraint analysis uses the requestors CAC, WSP-S, and ontologies to make the ‘best choice’.

The CAC is a file in XACML format which contains information about the requestor’s authorization relationship to organizations Web services. There are many possible relationships the requestor may have with the provider, for instance partner, client, anonymous invoker, etc. The CAC can be populated via SOAP messages containing authorization information in XACML format from Web service providers. The Web service providers can send the information to the requestor at account creation, during the formation of partnerships, when a service is invoked anonymously, and so on.

The WSP-S is an annotated WSP. As seen in figure 2 above, annotations can occur after the <All> tag in WSP. If there is one annotation for the entire WSP then it could be placed after the <ExactlyOne> tag. The first annotation in figure 2 describes authorization for a requestor whose role is *Inventory_Manager* for a *organization* that is a *partner* of the Web service provider. From the namespace it is seen that the concept *Inventory_Manager* is from the RBAC ontology and *Partner* is from the RosettaNet pips ontology.

Any domain specific ontology can be used for the annotations. However a quality of RBAC is that it has a structural hierarchy with relationships which lends itself to the creation of an ontology schema. The concepts of RBAC include organizational and professional roles. This fits well with the extension elements derived from the XACML representation of RBAC.

C. System Architecture

In this section, we present the architecture of our framework. The architecture is divided into three sections; Requestor Side,

Global Resources, and Service Side (Figure 3). The Requestor Side contains necessary information for discovery and authorization verification, Semantic Template and CAC respectively. A Semantic Template is created by a requestor to describe the function requirements of a Web service using ontological concepts [8].

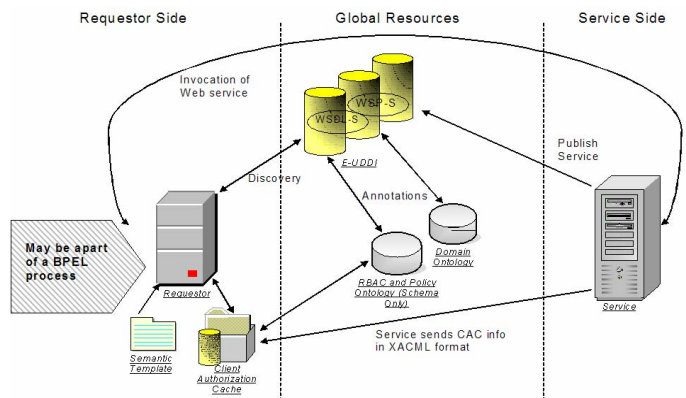


Figure 3 - System Architecture

Global Resources includes the E-UDDI for semantic discovery, the ontologies used for annotations, and the WS files. An E-UDDI is a UDDI registry that is capable of semantically matching the requirements in a Semantic Template with those in providers WSDL-S.

The Service Side contains the Web service and servers. Authentication and Authorization are implemented here and in our context are viewed as a ‘black box’. What is important is that the Service has described its authorization constraints in WSP-S using the RBAC ontology.

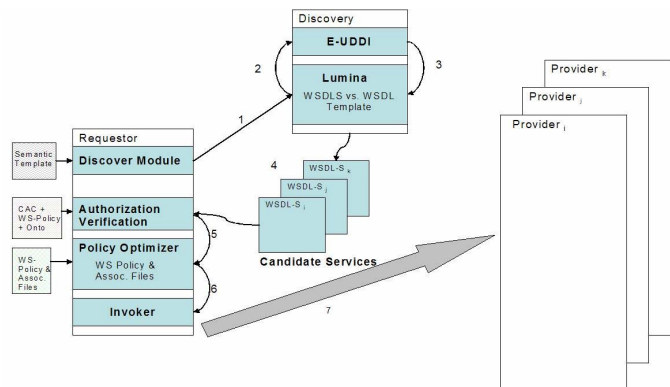


Figure 4 - Discovery Flow

D. Semantic Discovery Flow

As seen in figure 4, the requester sends the Semantic Template to the Semantic Discovery engine. In Steps 2 and 3, Lumina, a METEOR-S tool, works with a Semantic UDDI registry to discover candidate services. In step 4 these services are returned to the requester for Authorization Verification, constraint analysis. At step 5 of the 'Discovery Flow', a Policy Optimizer is used for further constraint analysis of the WS-Policy's functional and non-functional requirements. Lastly, step 7 invokes the service(s) of choice.

III. RELATED WORK

Much of the previous work on authorization for Semantic Web service is directed at implementing an access control enforcement structure [7]. [1] discusses the issues of a distributed heterogeneous network in which XACML and SAML are used for access control. Their approach converts XACML into SAML attributes for enforcement.

[5] uses attributes from credentials like SAML or Digital Certificates to make access control decisions in their implementation. While this can be done to some extent, these credentials were designed for authentication. Our approach is similar to [3]. They use ontologies to annotate OWL-S. It seems however that their approach adds another level of complexity to the adopted standards of WSDL and WSP.

A vision of a hybrid approach that incorporates real world concepts from an ontology with a rule based ontology is described in [3]. We agree with their conclusion of describing access control policies with multiple ontologies.

Our work differs from the previous works in several ways. We are extending the accepted standard WSP. We are using standards like RBAC and XACML in our annotation scheme and our ontologies. Our evolutionary approach builds on current standards where these other approaches are more revolutionary.

Policy matching in Semantic Web services is a complicated and relatively new area of research. [6] details an implementation of Semantic Policy matching using Semantic Web Rule Language. Their approach to matching Policies may be applicable to our approach for describing authorization.

[4] describes how to incorporate access control in a business process. Although it does not meet out dynamic authorization requirements, it illustrates fundamental capabilities in a workflow. This is relevant because Web services are the next generation of business processes.

IV. CONCLUSION AND FUTURE WORK

We have proposed a framework in this paper which will enable a requestor to Semantically discover a Web services which they have authorization to invoke. The research contributions of this paper are providing an annotation scheme for expressing authorization constraints for Web services.

In order for this framework to be utilized, there is some future work. The ontologies used for annotating must be

agreed upon by the Web services community. We believe that the NIST will play a large role in the development of an RBAC ontology, as well as RosettaNet in the development of business exchange ontology. The extensibility elements should be standardized by an internet authority. As should be a protocol for the exchange of authorization information between the provider and requestor is needed.

Future work on the protocol for the exchange of authorization information may be as simple as a Web service in which a requestor sends a Digital Certificate to a provider. The returned SOAP message is the authorization information in XACML format.

REFERENCES

- [1] G. López, Ó. Cánovas, A. Gómez-Skarmeta, S. Otenko, D. Chadwick; A Heterogeneous Network Access Service based on PERMIS and SAML; In Proceedings of 2nd EuroPKI Workshop, University of Kent, July 2005.
- [2] A. Toninelli, J. Bradshaw, L. Kagal, R. Montanari; Rule-based and Ontology-based Policies: Toward a Hybrid Approach to Control Agents in Pervasive Environments; Proceedings of the Semantic Web and Policy Workshop, International Semantic Web Conference, 7 November, 2005.
- [3] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, K. Sycara; Authorization and Privacy for Semantic Web Services; IEEE Intelligent Systems (Special Issue on Semantic Web Services), July 2004.
- [4] S. Wu, A. Sheth, J. Miller, Z Luo; Authorization and Access Control of Application Data in Workflow Systems; Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies (JIIS), Vol. 18, No. 1 (January 2002) pp. 71-94. Kluwer Academic Publishers.
- [5] S. Agarwal, B. Sprick, S. Wortmann; Credential Based Access Control for Semantic Web Services; http://www.aifb.uni-karlsruhe.de/WBS/sag/papers/Agarwal_Sprick_Wortmann-CredentialBasedAccessControlForSemanticWebServices-AAAI_SS_SWS-04.pdf.
- [6] K. Verma, R. Akkiraju, R Goodwin; Semantic Matching of Web Service Policies; Second International Workshop on Semantic and Dynamic Web Processes (SDWP 2005), Third International Conference on Web Services (ICWS'05), July, 2005.
- [7] M. Yague, A. Mana, J. Lopez, J. Troya; *Applying the Semantic Web Layers to Access Control*; 14th International Workshop on Database and Expert Systems Applications (DEXA'03)
- [8] K. Sivashanmugam, K. Verma, A. Sheth, J. Miller; *Adding Semantics to Web Services Standards*; Proceedings of the 1st International Conference on Web Services (ICWS'03), June 2003 pp. 395-401
- [9] et al Bob Atkinson, Giovanni Della-Libera; *Specification: Web Services Security (WS-Security)*; <ftp://www6.software.ibm.com/software/developer/library/ws-secure.pdf>; April 2002
- [10] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, D. Orchard; *Web Services Architecture*; <http://www.w3.org/TR/ws-arch/#security> Feb. 2004
- [11] et al Siddharth Bajaj, Don Box; *Web Services Policy Framework (WS-Policy)*; <ftp://www6.software.ibm.com/software/developer/library/ws-policy.pdf>
- [12] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, A. Sheth, K. Verma; *Web Service Semantics - WSDL-S*; <http://www.w3.org/2005/04/FSWS/Submissions/17/WSDL-S.htm>
- [13] D. Ferraiolo, R. Sandhu,
- [14] S. Gavrilu, D. Kuhn, R. Chandramouli; *Proposed NIST Standard for Role-Based Access Control*; <http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf>
- [15] et al Tim Moses; *eXtensible Access Control Markup Language (XACML) Version 2.0*; OASIS Standard Feb. 2005; http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

