Implementing a Secure Annotation Service

Imran Khan, Ronald Schroeter, and Jane Hunter

The School of ITEE
The University of Queensland,
St Lucia, Queensland, Australia
{imrank, ronalds, jane}@itee.uq.edu.au

Abstract. Annotation systems enable “value-adding” to digital resources by the attachment of additional data in the form of comments, explanations, references, reviews and other types of external, subjective remarks. They facilitate group discourse and capture collective intelligence by enabling communities to attach and share their views on particular data and documents accessible over the Web. Annotation systems vary greatly with regard to the types of content they can annotate, the extent of collaboration and sharing they allow and the communities which they serve. However many applications share the need to authenticate the source of annotations and restrict access to them - in order to protect intellectual property rights or personal privacy. This paper describes a secure, open source annotation system that we have developed that uses Shibboleth [1] and XACML [2] to identify and authenticate users and restrict access to annotations stored on an Annotea [3] server.

1 Introduction

Annotations have long been used as a tool to facilitate collaborative scholarly discourse. They enable users to attach additional material such as comments, notes, queries, assessments, references to resources such as documents, images or datasets. When applied to digital resources shared via the Web, they provide a very powerful collaborative tool.

Currently available annotation systems vary widely with respect to the types of content they annotate, the extent of collaboration and sharing they allow and the communities which they serve [4]. Although they have been successfully applied to domains including education, research, medicine [5] and neuroscience [6] in order to capture and exchange metadata, ideas, opinions and interpretations, evaluation of these applications indicates limitations in existing commercial and prototype systems. Current systems are limited by: lack of responsiveness, use of non-standard proprietary technologies; lack of authentication of the annotations’ creator; limited search capabilities; lack of security mechanisms; inability to reply to/stagger annotations; asynchronous sharing only; support for limited media types; coarse granularity and unstructured annotations (single field, free text only).

The main focus of the work described in this paper is to provide annotation tools for collaborators within eResearch. A critical requirement for such a domain is the need to be able to restrict access to annotations attached to a particular collection of digital resources - to a particular group of trusted colleagues - for reasons of privacy,

confidentiality or protection of intellectual property. This is particularly important within eScience, where the annotation or interpretation of the raw document or data, is often more valuable than the target of the annotation. Also by providing researchers with a robust, reliable security infrastructure, they may be more willing to engage in the exchange of views and ideas – a key to successful inter-organizational collaboration.

The security requirements for annotations involve two levels of protection:

- protecting the annotation server through identity management and authentication;
- protecting individual annotations through the specification of access policies that define permissible types of access (e.g, list, read, delete) by individual users or user groups based on user attributes.

Within this paper we describe an open source implementation of a secure annotation service that we have developed. Our implementation involves the combination and extension of a number of existing open source technologies that are based on open standards:

- Annotea – a Web-based annotation server developed by the W3C [7];
- Shibboleth – an Internet2 middleware initiative that enables identity management and secure access to Web resources shared amongst multiple organizations [1];
- XACML (eXtensible Access Control Markup Language) – XML-based language for defining and enforcing access control policies [8].

The remainder of this paper describes in detail the secure annotation system that we have built. Section 2 describes previous related activities in the development of annotation systems and security mechanisms. Section 3 describes the overall architecture of our system and its main components. Section 4 illustrates the user interface and system functionality. Section 5 provides an evaluation of the system and describes future work. Section 6 provides a brief conclusion.

2 Background and Previous Work

Significant prior work has been carried out on both web-based annotation systems and on identity management and role-based access control. Rather than re-invent the wheel, we carried out an analysis of existing systems to determine if any currently available solutions satisfied our needs and hence could be integrated, refined or extended.

2.1 Existing Annotation Systems and Annotea

Fig. 1. Annotea's Annotation Schema [5]

A survey of current Web-based annotation systems [4] reveals that they vary in the way in which annotations may be attached, the way in which they are presented and in the access control mechanisms. None of the surveyed systems provide the kinds of fine-grained access control

mechanisms that is required by collaborative teams of scientists engaging in eResearch.

Through earlier work [9] we identified Annotea [7] as an ideal approach for implementing an annotation server. Annotea is a Web-based annotation system that uses Resource Description Framework (RDF) [10] to model annotations as a set of statements or assertions. These annotations are stored in a HTTP enabled server, which enables clients Annotea enabled clients [11] [12] to query, update, post, delete and reply to annotations. Currently there are two publicly available implementations e Annotea servers: Zope [9] and W3C PerlLib[13]. Figure 1 illustrates the RDF annotation schema used to describe various properties of an annotation including its author, title, date of creation, body and context. Annotea's use of open W3C standards such as RDF, XPointer, XLink and HTTP makes it possible to easily adapt or extend the scheme. Annotea can also be easily extended to allow for the annotation of media types other than text. For example Vannotea [14] extends Annotea to enable the annotation of videos. Vannotea and applications similar to it clearly identified the need to further extend the Annotea to enable fine-grained access control to annotations [15].

2.2 Identity Management and Shibboleth

Harris et. al. generated a comprehensive report describing access management (AM) systems used in the UK Higher Education sector [16]. The most prominent systems identified included: Microsoft's Passport, Liberty Alliance, WS-Security, PAPI, Athens and Shibboleth. Of the six systems, only three are targeted at the higher education domain, while the other three (Passport, Liberty Alliance and WS-Security) are primarily focused on business-centric solutions. Morgan et al [17] describe Shibboleth as "an open-source system that extends Web-based applications and identity management for secure access to resources among multiple organizations." Shibboleth is based upon a number of open standards including HTTP, XML, XML Schema, XML Signature, SOAP (Simple Object Access Protocol). In particular it is dependent on:

- SAML [18] Security Assertion Markup Language for the exchange of assertions between the Identity Providers and Service Providers
- eduPerson [19] – an EDUCAUSE initiative to define a standard set of person attributes in higher education environments

2.3 XACML

As Lorch et al [8] explains, Shibboleth lacks a dynamic and distributed approach to access control. XACML enables us to address these issues. XACML (eXtensible Access Control Markup language) is an XML-based language used to describe general purpose access control policies and also an access control decision request/response language [18]. XACML also specifies the structure and syntax of the requests and responses. Requests are composed of attributes associated with the requestor, resource being accessed and the action being performed.

repository. Objects are linked to particular policies through their policy property – which is specified by a URL. This approach has the benefit of enabling multiple annotations to use the same policy. If a policy is modified, the changes will effect all those annotations associated with the policy.

3.1.2 XACML Module and Policies

This module is responsible for implementing the *Role Based Access Control* functionality and is based upon Sun’s XACML implementation [2]. It makes decisions on whether a particular request is permitted based upon the role/attributes of the person making the request. There are three types of actions permissible on annotations by users other than the creator:

- LIST – viewing of annotation metadata (e.g., author, creation date, etc.)
- READ – viewing of the annotation body.
- READ_POLICY – viewing of the annotation policy.

Figure 4 illustrates an example policy and request. Each **Policy** consists of a set of **Rules** related to whether a specific operation is permitted by a particular Subject. The Subject is described by a set of attributes which identify the credentials of a particular user e.g. affiliation, role. In Figure 4a, members of the Staff “Group” have an attribute *eduPersonAffiliation* equal to “staff”. In the example, staff are permitted to perform all three operations on annotations whilst students are denied access to all three. Given the example policy in Figure 4a, the example request (in Figure 4b) - that a student to be permitted to read policy 123, will be denied. It is important to note that although XACML policies provide much more expressiveness than we provide, we have deliberately kept the user interface (Figure 6) simple so that end users can create policies themselves.

Fig. 4. (a) Example Policy and (b) Example Request

The XACML module is implemented through three steps. The first step involves gathering attributes about the requester from the requestor’s IDP through Shibboleth’s SAML assertions. Using these attributes, an XACML request is created – it specifies the action to be performed, the resource requested and the attributes of the requestor. The second step involves locating the policy associated with the resource being requested by querying the RDF repository for a policy with a given URL. Thirdly the

retrieved policy is compared with the request and Sun's XACML implementation generates an XACML response specifying whether the request is permitted or denied.

3.1.3 Shibboleth (SAML) Attribute Assertions

The annotation server depends on Shibboleth to provide the necessary eduPerson attributes about a requestor. These attributes are used by the XACML module to make an access control decision. Shibboleth itself is a complex architecture and details are available from [20]. Each site within a Shibboleth federation consists of either/both an origin (identity provider) and target (service provider). In terms of our annotation system, a user's attributes are provided by their origin, which stores them in an LDAP server. The annotation server is hosted on the target site and is accessible to members of organizations that are part of the federation and have sufficient access privileges to the annotation server.

3.1.4 Jena Database

Jena [21] provides an API to an RDF repository and in the context of this system is responsible for enabling the storage and interfacing of data – including annotations, policies and annotation bodies. The Jena API also enables us to search the annotations - via the creator, date, language and in_reply_to fields.

3.2 Client-Side

The user's client side application is responsible for the user interface that enables: the retrieval and display of annotated web resources; display, search and browsing of annotations the creation, editing, deletion and attachment of annotations to Web resources; the creation, editing and attachment of access policies to annotations.

Although a number of client-side annotation tools exist for annotating Web resources (Amaya [12], Annozilla [11], and Vannotea [14]) none of these provide an interface suitable for specifying XACML access policies. Consequently we developed our own client-side application using .NET to allow the display and editing of annotations and policies.

4 The User Interface

For testing and illustrative purposes, we used the ePrints archive at the University of Queensland. Figure 5 shows the user interface after an authenticated user accesses the annotation server and retrieves a particular annotated publication. The annotations are displayed in the top left-hand frame, the details of a selected annotation are in the bottom left-hand frame and the publication is displayed in the right hand frame.

Figure 5 also illustrates the user interface for creating and attaching an annotation. We have extended Annotea to support structured annotations that contain a number of fields including hyperlinks, files, free text or controlled vocabularies.

Fig. 5. User Interface showing Sidebar with threaded replies and dialog box for creating annotations

Fig. 6. User Interface for defining Policy (a) Groups and (b) Rules

Figure 7 shows the user interface developed for defining groups and policies. Firstly it enables ‘Groups’ to be defined by sets of common attribute values. In Figure 8a, the group *uq_members* is defined as users with (*eduPersonAffiliation* = *staff*, *eduPersonOrgUnitDN* = *dke* and *eduPersonOrgDN* = *itee*) where the attributes are issued by *uq.edu.au*. The policy in Figure 9a has three groups - *uq_members*, *monash_members* and *jcu_members*. The second part of policy definition involves defining access rules for each of the groups. In Figure 10b, the Group *jcu_members*

are permitted to *List* and *Read* annotations, but not *Read Policy*. This interface makes it easy for users to define new groups, modify/remove existing groups and define/modify policies.

5 System Evaluation and Future Work

5.1 System Evaluation

To date, system evaluation has comprised thorough unit and system testing. This involved testing the creation, editing and deletion of policies and annotations. We also tested policy enforcement by logging on as users with different attributes and modifying attributes directly in the LDAP directory. In all cases the annotation server behaved as expected. However the testing phase did reveal some problematic issues. These included:

- Allowing the deletion and update of annotations can lead to ‘hanging references’ where replies refer to annotations which have been updated or deleted.
- The use of URLs to identify policies enables them to be re-used and applied to multiple annotations. However this may cause problems when a policy referred to by multiple annotations is updated/deleted.

5.2 Future Work

Aspects of this work that would benefit from further investigation include:

- Thorough user evaluation: detailed usability studies are required to acquire user feedback and determine functional requirements of user groups as well as improve, refine and extend the system.
- Reduce reliance on Shibboleth: approaches other than Shibboleth will enable the annotation server to be used outside of the Higher Ed sector.
- Annotation of PDF files and spreadsheets: the popularity of publishing scholarly information in PDF format and storing scientific data in spreadsheets indicates an increasing demand to be able to annotate data in proprietary formats.
- Access policies based on document attributes: it would be interesting to investigate policies that are based on attributes of the digital resources or their annotations.
- Complex querying: the integration of SPARQL to allow more complex queries over the annotation server while enforcing access constraints.
- Post-processing of annotations: currently authors are notified via RSS when replies to their annotations are made. Other examples include the Multivalent Browser [22] which can review and incorporate suggested changes within documents.
- Ontology-based annotations or the annotation metadata could be used to process annotations and automatically classify/rank annotations and resources.
- Scalability: further investigation is required to determine how the system performs as the number of annotations, access policies and users grows.

6 Conclusions

This paper describes a secure annotation service that we have developed by combining and extending a number of existing open source technologies. Secure, trusted annotation servers are required in many domains including telemedicine, higher education and collaborative eResearch. By providing clinicians and researchers with the necessary support for authenticating the source and protecting the confidentiality and intellectual property of their annotations, they will be more willing to share their views and engage in inter-organizational collaborations with trusted colleagues. Moreover, the modular design and interoperable technologies that we have adopted, makes it easy to quickly adapt the server to a variety of different media types, different domains and different communities.

References

- [1] Internet2, "Shibboleth Project," 2005, <http://shibboleth.internet2.edu/>.
- [2] S. Proctor, Sun Microsystems, "XACML API," 2004, <http://sunxacml.sourceforge.net/>.
- [3] M. Koivunen et al, "Annotea: an open RDF infrastructure for shared Web annotations," in *Proceedings of the 10th Intl conference on World Wide Web*, Hong Kong, ACM Press, 2001
- [4] R. Heck et al, Department of Mathematics and Computer Science, Grinnell College, "A Survey of Web Annotation Systems," 1999, http://www.math.grin.edu/~rebelsky/Annotations/Summer1999/Papers/survey_paper.html.
- [5] M. Lewkowicz et al, "A Web-based Annotation System for Improving Cooperation in a Care Network," in *ICWE Workshops*, pp. 227-239, 2004.
- [6] M. Gertz et al, "Annotating Scientific Images: A Concept-Based Approach," in *14th Intl Conference on Scientific and Statistical Database Management* 2002.
- [7] R. Swick et al, W3C, "Annotea Protocols," 2002, <http://www.w3.org/2002/12/AnnoteaProtocol-20021219>.
- [8] M. Lorch et al, "First experiences using XACML for access control in distributed systems," *Proc of the 2003 ACM workshop on XML security* Fairfax, Virginia ACM Press, 2003.
- [9] Zope, "Zope Annotation Server," 2005, <http://www.zope.org/Members/Crouton/ZAnnot/>.
- [10] D. Brickley and R. V. Guha, W3C, "Resource Description Framework (RDF) Schema Specification 1.0," 2005, <http://www.w3.org/TR/2000/CR-rdf-schema-20000327>.
- [11] M. Wilson, Mozdev.org, "Annozilla (Annotea on Mozilla)," 2000, <http://mozdev.org/>.
- [12] I. Vatton, W3C, "Amaya," 1994, <http://www.w3.org/Amaya/>.
- [13] W3C, "PerlLib Annotations Server HOWTO," <http://www.w3.org/1999/02/26-modules/User/Annotations-HOWTO>.
- [14] R. Schroeter et al, "Vannotea -A Collaborative Video Indexing , Annotation and Discussion System For Broadband Networks," in *Knowledge Markup and Semantic Annotation Workshop, K-CAP 2003*, Sanibel, Florida 2003.
- [15] J. Hunter et al, "Using the Semantic Grid to Build Bridges between Museums and Indigenous Communities," in *Semantic Grid Applications Workshop*, Honolulu 2004.
- [16] N. Harris et al, "Access Management Report," London School of Economics 2002.

- [17] R. L. Morgan, S. Cantor, W. Hoehn, and K. Klingenstein, "Federated Security: The Shibboleth Approach," *Educase Quarterly*, vol. 27, pp. 12-17, 2004.
- [18] R. Cover, Oasis, "Cover Pages: Security Assertion Markup Language (SAML)," 2005, <http://xml.coverpages.org/saml.html>.
- [19] Directory Working Group (MACE-Dir), Internet2 Middleware Architecture Committee for Education, "EduPerson Object Class Specification (Draft)," 2006, <http://www.nmi-edit.org/eduPerson/draft-internet2-mace-dir-eduperson-latest.html>.
- [20] T. Scavo and S. Cantor, Internet2, "Shibboleth Architecture," 2005, <http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-conformance-latest.pdf>.
- [21] B. McBride, "Jena: A Semantic Web Toolkit," *IEEE Internet Computing*, vol. 6, pp. 55-59, 2002.
- [22] A. P. Thomas et al, "The multivalent browser: a platform for new ideas," in *Proceedings of the 2001 ACM Symposium on Document engineering*, Atlanta, Georgia, USA, ACM Press, 2001.