USING SAML TO LINK THE GLOBUS TOOLKIT TO THE PERMIS AUTHORISATION INFRASTRUCTURE

Authors David Chadwick[1], Sassa Otenko[1], Von Welch[2]

Affiliation [1]ISI, University of Salford, Salford, M5 4WT, England. [2]Argonne National Laboratory, University of Chicago

Abstract: In this article the new trend in authorisation decision making will be described, using the Security Assertions Mark up Language (SAML). We then present an overview of the Globus Toolkit (GT), used in Grid computing environments, and highlight its authorisation requirements. We then introduce the PERMIS authorisation infrastructure and describe how it has been adapted to support SAML so that it can be deployed to make authorisation decisions for GTversion 3.3.

Key words: SAML, Grid Computing, Authorisation, X.509 attribute certificates, RBAC, policy decision making, PDP, PEP.

1. INTRODUCTION

The Security Assertions Markup Language (SAML) [1] has been designed by the Organization for the Advancement of Structured Information Standards (OASIS) to provide a universal mechanism for conveying security-related information between the various parts of an access control system. It is an XML-based language for encoding security request and response messages between the initiator of an access request, the authentication service, the authoriser (termed an attribute authority) and the access control decision function (ADF). Some of these parts of an access control system may be grouped together, in which case they will not need to send SAML messages between themselves, and may use some sort of API to convey the necessary information between each other.

The Globus Toolkit (GT) is an implementation of Grid software, which has a number of tools that make development and deployment of Grid Services easier [2]. One of the key features of this toolkit is secure communications. However, Globus Toolkit has limited authorisation capabilities based on simple access control lists. To improve its authorization capabilities a SAML authorisation callout has been added. The important consequence of this is that it will be possible to deploy an authorisation service that the GT will contact to make authorisation decisions about what methods can be executed by a given client. One such authorisation service is PERMIS [3]. Whilst the original PERMIS Java API was intended for local calls only, and didn't have any network interface, a PERMIS Authorisation Service has been developed to provide authorisation decisions for the Globus Toolkit through the SAML callout.

2. OVERVIEW OF EXISTING TECHNOLOGY

2.1 SAML

SAML is a language for expressing security-related information. It defines message formats in XML for Queries and Responses. It also defines a request-response protocol in SOAP over Http for carrying the SAML messages. SAML Queries are sent to a decision-making service whilst Responses, in the form of SAML Assertions, are returned. These assertions can then be coupled with a further Query and sent to other decision making services to aid them in their own decisions. In the SAML model there are three decision-making services: the Authentication decision-making service, the Attribute decision-making service and the Authorisation decision-making service (see Figure 1). Each decision-making service uses its associated policy and the user's credentials to evaluate the Query. After the SAML Query has been evaluated, a SAML Response is generated and this may be forwarded to another decision-making service, until it finally reaches the Policy Enforcement Point (PEP) of the application, which will determine the ultimate fate of the user's application request. The PEP is equivalent to the Access control Enforcement Function (AEF) in ISO 10181-3 Authorisation Framework [4].

SAML does not mandate any exact sequence of message flows for access control decision making. However, a typical flow might be as follows. A user's access request is presented to a PEP/AEF, and comprises the user's name, the user's credentials, the target to be accessed and the requested mode of access. The PEP could then sequentially present portions of this request to the three decision-making services. Firstly the user's name and credentials are presented to the Authentication Authority, which confirms the identity of the user. Next the authenticated name of the user (or the authentication assertion returned by the Authentication Authority) is presented to the Attribute Authority, which confirms the assignment of certain attributes to the user. Finally the attribute assertions, the name of the target and the requested mode of access are presented to the PDP, which makes an access control decision. The PEP then acts on this decision and either forwards the user's request to the target (if the PDP granted the request) or returns an error message to the user (if the PDP denied the request).

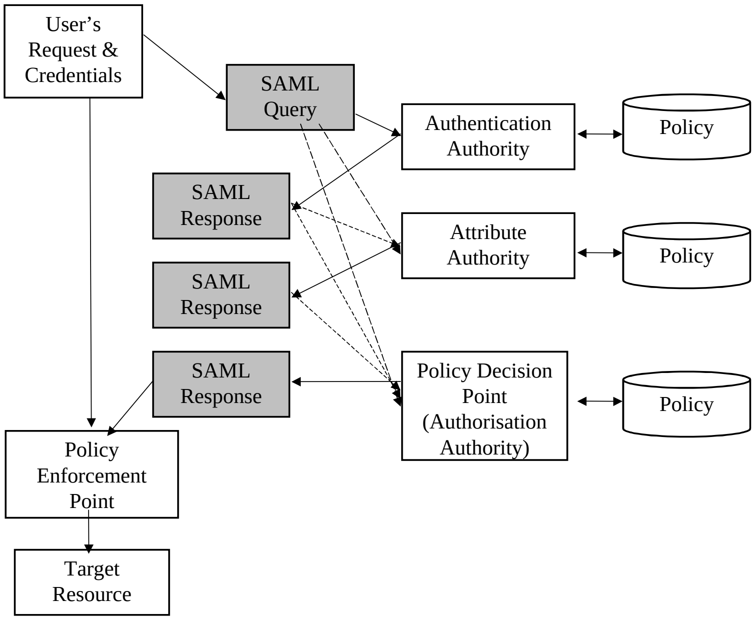

Figure 1. SAML

The SAML messages can be digitally signed, which makes them tamperproof, i.e. the messages can be sent as plaintext across untrusted networks. Alternatively, the SAML protocol messages could be sent as SOAP over Http over SSL, which can also protect them from eavesdropping.

Until quite recently most uses of SAML were limited to authentication and attribute usage e.g. as in Shibboleth [5]. Authorisation decisions were usually made locally either based on the user's identity (in Access Control Lists) or on the attributes/roles of the user (in simple scripts).

Comment [VW1]: Shibboleth is a SAML attribute authority. I think we want to delete this paragraph.

As virtual communities and Grid computing started to develop, identity-based systems for authorization became increasingly difficult to manage due to the distributed nature of the user communities. To accommodate these communities, new authorisation systems were required that would make the decisions based on the attributes of the initiator rather than their identity. Another prerequisite for SAML authorisation messaging was that there should be a centralised decision-making point for a number of remote services, governed by the same policy. Thus Community Authorisation Server (CAS) [6], Akenti [7], PERMIS and others started to appear.

2.2 The Globus toolkit

The Globus Toolkit (GT) is a set of tools for building Grids that includes tools for resource discovery, job submission and data movement. Version 3 of the Globus Toolkit (GT3), includes support for Grid Services based on the Open Grid Service Infrastructure (OGSI) standard, which defines extensions to Web Services for lifetime management and stateful instances (among other things outside the scope of this paper). GT3 provides a Grid Service Container to host Grid Services instances, which provides services such as message marshalling/de-marshalling, authentication and authorization.

A virtual organization (VO) is a collection of users and resources, distributed across a number of geographic and administrative domains, which share common policies for access control. Initially access control was solved in GT through the use of simple access control lists called grid-map files, which performed mapping of access rights based on the user's identity. Such simplistic policies were robust, but failed to scale as the VOs grew in size and spanned larger numbers of institutions. To provide more flexible authorisation solutions, it was decided to provide a SAML authorisation callout in GT3 to allow the use of advanced authorization services. The effect of this is that the Grid Service Container would be able to contact the centralised Policy Decision Point to make access control decisions for invocations of services it hosts. In this design the PDP becomes yet another Grid Service, which provides authorization decisions through a standard message format i.e. SAML.

Now it is possible to create a CAS, Akenti, or PERMIS port that would make access control decisions for Grid Services, based on queries and decisions in the form of SAML Queries and Responses which are enforced by the Grid Service Container. A detailed description of the operation scenario is given later, with the example related to PERMIS.

2.3 PERMIS

PERMIS is a policy based authorisation infrastructure, in which a user is granted rights to access a resource based on the authorisation policy for the resource, and the set of attributes (or roles) that the user possesses. A user's attributes are stored in digitally signed X.509 attribute certificates [10], and are allocated by the authorities in charge of the various attributes. Thus a "doctor" role attribute could be allocated by the General Medical Council, whilst a "project manager" role attribute could be allocated by the head of a department. A "date of birth" attribute could be allocated by a national registrar. These attribute certificates are then stored in various LDAP directories.

PERMIS was designed with a Java API between the PDP and PEP providing the access control decisions. Given the name of the user, it retrieves the user's attributes/roles and makes decisions based on them. The authorisation policy, written in XML, expresses which users can be assigned what attributes/roles by whom, and what privileges are bound to each of the attributes/roles. The XML policy is then inserted in an X.509 attribute certificate, signed by the manager who wrote it, and stored in an entry in an LDAP server.

When an application starts up, its PEP/AEF passes to the PERMIS ADF/PDP the name of the manager, the location of the LDAP directory, and the unique number of the policy to be used (each policy is assigned a globally unique number – actually an object identifier [12] – so that a manager can create different policies to be used in different contexts). The PERMIS ADF retrieves the policy X.509 AC from the LDAP directory, checks the signature and policy number, and if both are correct, uses this policy for its decision making.

3. THE IMPLEMENTATION

3.1 Extensions to SAML

A standard SAML Response contains a complete list of all the allowed actions which were contained in the SAML Query. While this is useful in cases where the response is passed to a third party, in the case where the query was generated by the consumer of the response it can introduce unnecessary overhead. In these cases the consumer of the response must parse the entire list of actions, when it may only be interested in a "yes" or "no" answer regarding the entire list as a whole.

For the sake of performance, new SAML Requests and Responses have been proposed – they are shorter and more concise versions of the standard SAML Authorization Decision Request and Authorization Decision Statement (passed inside a SAML Query and SAML Response respectively).

These new messages allow the query to request, and the decision to contain, a simple yes/no response to all the actions contained in the query. This allows the authorization service to easily encode, and the Grid Service Container to easy parse, the response. The specification of these new SAML messages has been written by the Global Grid Forum's (GGF) Open Grid Service Architecture Authorisation working group (OGSA-AuthZ) [9].

3.2 Extensions to PERMIS

As the demand for an authorisation service appeared in the Globus Toolkit, the PERMIS researchers closely collaborated with the GT team to provide practical input into the design of the SAML messages to be used in GT3. In parallel a standalone PERMIS ADF was developed. Whilst the original PERMIS RBAC ADF had to be contacted via its Java API, the standalone ADF has a networking interface, through which it can receive SAML Queries and send back SAML Responses. In addition to this, the PERMIS ADF can pose as a Grid Service, so it can be used as a centralised PDP of a Grid application. It can also easily be embedded as a Service Authorisation – i.e. a PERMIS Authorisation "Callout" can replace the GT3 standard SAML Authorization Callout, and make local decisions for local Grid Services.

The original PERMIS architecture supported only LDAP Distinguished Names as Target identifiers. This allowed PERMIS to group targets into domains for easier expression of the policy. It was noticed during the early development stages of PERMIS that in fact people use various kinds of naming conventions for identifying targets, e.g. IP addresses, DNS names, URIs etc., so the design of the PERMIS API incorporated Principals, as an abstraction for such identifiers. However, the original PERMIS pilot sites did not have any specific requirements about which type of naming to use, and so for consistency purposes target naming was chosen to be the same as subject naming. Since subjects were named using X.500/LDAP distinguished names, then so too were targets.

When porting the code to work with the Globus Toolkit, it became necessary to allow other kinds of naming conventions to be used for targets, specifically because in Globus the intended targets are Grid Services, and they already have non-LDAP identifiers in the form of Grid Service Handles (GSHs), encoded in the form of a URI. URIs are hierarchical names like LDAP DNs, and this helps to group targets into domains (although of course in any particular Grid application the relevant targets at different sites may have totally unrelated URIs). URIs do not provide any further refinements for targets, unlike LDAP, which provides Object Classes to help to further distinguish between the different kinds of target e.g. printers or cpu clusters. This is one example of identifying targets by their attributes as well as or instead of their names.

Besides changing the ADF interface to support URI target names, the PERMIS policy syntax also had to be extended to support URIs as Target Identifiers[1]. This allows the manager to specify target names as URIs.

The PERMIS ADF assumes the subjects are authenticated, but it can recognise unauthenticated subjects, and now it will return public access rights to such subject names, i.e. it will grant access to the targets that do not require any roles to perform certain actions.

The Java code for constructing subject and target domains is the same, so in principle subjects could also be identified by their URIs. This would allow, for example, particular Grid Services (which are the targets for normal Grid users) to act as subjects and to make requests to other services protected by the PERMIS infrastructure. For example, a Grid service could make a request to an attribute repository protected by the PERMIS ADF, and PERMIS could decide if the particular Grid Service is allowed to retrieve certain user Attribute Certificates (specified as targets), thus enforcing a user's Privacy Policy.

The question of how to locate the credentials of a subject identified by a URL in a repository that uses LDAP naming can in fact be solved in at least two ways. The first and easiest way is to use URLs that conform to RFC 2255 [11]. This specifies how LDAP URLs can be used to retrieve information from LDAP repositories. An alternative way is to embed the URL as the latter part of the LDAP Distinguished Name and to configure the LDAP repository with the prefix DN.

For file-based repositories, e.g. Web-servers or file-servers, it is possible to construct filenames out of the subject's identifier, e.g. an MD5 hash of the normalised subject identifier (either the LDAP DN or URL) can be used to locate the files containing the necessary credentials[2].

The extended SAML Requests designed by the OGSA-Authz group and specified in [9] may contain Attribute Reference elements. In essence these are repository URLs from where the subject's attributes should be retrieved by the Authorisation Service. This is a “semi-push” or “controlled pull” model, i.e. the subject doesn't have to push all the credentials to the Authorisation Service as SAML evidence, or rely on the Authorisation Service to pull whichever attributes it wants from where, but instead can provide a reference to the repositories that contain them. The Authorisation

[1] In fact, any URI can be used, but a specific URI handler must be registered with the PERMIS RBAC at initialisation time.

[2] For example, this is the way some Public Key Certificates are located on some web-servers.

Service will then pull the attributes from this referenced repository. Note that where the attributes are stored as digitally signed X.509 ACs, they are tamperproof and so it is still possible to use such references for making secure decisions. To cater for this extension to SAML, a new parameter to the PERMIS getCreds method has been introduced. It provides the PERMIS Authorisation Service with a list of repositories to contact to get the subject's credentials.

To configure the PERMIS Authorisation Service at initialisation time, it is necessary to specify the URL of the LDAP repository where the policy is located, the Distinguished Name of the manager issuing the policy, and the Object Identifier of the actual policy to be used. These parameters are specified in the GT3 Service Container deployment descriptor. Unfortunately, there is currently no way to make the deployment descriptor tamperproof. Therefore to ensure that the Grid service is always correctly configured, it is recommended that a human security officer should always be present at the service start-up time, to check that the configuration parameters that the service uses are the expected ones.

3.3 Operation Scenario

There are two modes of interaction between GT3 and PERMIS Authorisation. One mode is remote, the other mode is local. In remote mode a PERMIS Authorisation Service is set up to serve a number of Grid Service Containers. In local mode of operation each Grid Service Container has its own PERMIS Authorisation set up as an Authorisation Handler. In local mode there are no SAML messages, and authorisation is done via the PERMIS API, so only a comparison of this mode to the remote mode of operation is given in section 4. The remote mode of operation is described below.

When a subject makes a request of a Grid Service, the subject is typically authenticated by the Grid Service Container using SSL and the user's X.509 certificate (see [8] for details). The subject may also invoke operations anonymously, in which case a special identifier (*) is used to indicate an anonymous user. The service container generates a SAML Authorisation Request, which includes an identifier of the subject, an identifier of the service (its Grid Service Handle, a URI), and the name of the operation being invoked. This information is enveloped in a message containing a timestamp and signature along with other information to protect the message from tampering and prevent replay attacks. This message is sent to the trusted Authorisation Service as defined in the service's configuration, e.g. a PERMIS Authorisation Service.

The Authorization Service parses the request, uses the policy to make an authorisation decision about the request, and returns a response containing the decision (again enveloped in a message which includes a signature and replay protection). Only an affirmative decision will cause the service container to allow for the action requested by the user to be executed.

4. DISCUSSION

The implementation described above should help to provide Role-Based Access Controls in Virtual Organisations built on Grids. Each of the resource providers will write a PERMIS policy for resource usage, and authorise collaborative institutions to issue roles to their members. The collaborative institutions will issue role assignment X.509 Attribute Certificates to their members, and based on these and the policy, the PERMIS ADF will make the authorisation decisions.

The implementation is now complete and pilot testing is due to take place during the next 5 months, so that actual results should be present in time for the CMS2004 conference.

It is still questionable how efficient it is to have such an authorisation service called via SAML/SOAP/Http rather than to have a PDP/ADF called locally via a programmable API. The gain that can be achieved using the centralised PDP is that in single sign-on distributed systems such as the Grid, the authorisation tokens (attribute certificates) of the user would have to be retrieved only once, rather than at each resource of the distributed system. In most cases this might give a doubtful gain in performance because SAML messages still have to be generated for each request.

A centralised PDP should make policy management easier – security managers do not have to change the policy at each PDP. However, the PERMIS infrastructure has already addressed this problem by storing its policy as a digitally signed AC in a central LDAP repository, from where all the distributed systems can retrieve the same policy.

A centralised PDP can provide more user privacy. For example, it is easier to conceal the user’s identity in a single trusted PDP (and use a pseudonym throughout the rest of the system), rather than spread this knowledge across PDPs at each resource site.

A centralised PDP makes implementation of the Principle of Separation of Duties much easier to enforce – it is easier to track what roles a user has assumed in the past, so his further requests do not clash (e.g. the Payment Requestor cannot be a Payment Guarantee for the same order, and an

Accountant cannot be an Auditor for the same transaction). This is much more difficult to enforce with multiple distributed PDPs.

Having said all this, it should be noted that most authorisation decision systems today are local, i.e. no centralised decision-making is done. Decentralising the decision-making process has its benefits, which are usually connected with the speed of decision making, and the up-to-date reflection of the system's state in the PDP (as contextual parameters).

The Grid environment encourages institutions to collaborate with each other. The links between these institutions may be established in a fairly spontaneous way, and these institutions may already have their own Privilege Management Infrastructures in place. This means that the participating institutions may have already assigned roles to their members. It is important in this case that the collaborating institutions are able to recognise each other's role assignments and optimally to be able to compare the roles issued by the different participating institutions. Currently the PERMIS policy has to be configured with all the different roles, and permissions assigned to each. In the future we expect to be able to express role mappings, and one of our ongoing projects aims to facilitate dynamic cross-institutional virtual organisations using existing PMIs.

5. REFERENCES

[1] OASIS. "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)". 19 April 2002. See http://www.oasis-open.org/committees/security/

[2] Globus toolkit, http://www.globus.org/toolkit

[3] D.W.Chadwick, A. Otenko, E.Ball. "Implementing Role Based Access Controls Using X.509 Attribute Certificates", IEEE Internet Computing, March-April 2003, pp. 62-69.

[4] ITU-T Rec X.812 (1995) | ISO/IEC 10181-3:1996 "Security Frameworks for open systems: Access control framework"

[5] Shibboleth Project, available at http://shibboleth.internet2.edu/

[6] L. Pearlman, V. Welch, I. Foster, C. Kesselman, S. Tuecke. "A Community Authorization Service for Group Collaboration"**.** Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[7] Johnston, W., Mudumbai, S., Thompson, M. "Authorization and Attribute Certificates for Widely Distributed Access Control," IEEE 7th Int Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE), Stanford, CA. June, 1998. Page(s): 340 -345 (see also http://www-itg.lbl.gov/security/Akenti/)

[8] Von Welch, Frank Siebenlist, Ian Foster, John Bresnahan, Karl Czajkowski, Jarek Gawor, Carl Kesselman, Sam Meder, Laura Pearlman, and Steven Tuecke. Security for grid services. In *Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*. IEEE Computer Society Press, 2003.

[9] Von Welch, Frank Siebenlist, David Chadwick, Sam Meder, Laura Pearlman. "Use of SAML for OGSA Authorization", Jan 2004, Available from https://forge.gridforum.org/projects/ogsa-authz

[10] ISO 9594-8/ITU-T Rec. X.509 (2001) The Directory: Public-key and attribute certificate frameworks

[11] T. Howes, M. Smith. "The LDAP URL Format", RFC 2255, Dec 1997

[12] ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:1998, Information Technology - Abstract Syntax Notation One (ASN.1): Specification of Basic Notation