Ontology Based Interoperation for Securely Shared Services

Security Concept Matching for Authorization Policy Interoperability

Ioana Ciuciu1*, Gang Zhao1, David W Chadwick2**, Quentin Reul1, Robert Meersman1, Cristian Vasquez1, Mark Hibbert2, Sandra Winfield3*** and Thomas Kirkham3

1 STARLab, Vrije Universiteit Brussel, Brussels, Belgium
2 School of Computing, University of Kent, Canterbury, UK
3 University of Nottingham, Nottingham, UK
* iciuciu@vub.ac.be, ** d.w.chadwick@kent.ac.uk, *** sandra.winfield@nottingham.ac.uk

Abstract—This paper addresses the problem of access control in the context of unified distributed architectures, in which a local authorization policy is not able to recognize all the terms applicable to the authorization decision requests. The approach is based on semantic interoperability between the different services of the architecture. More specifically, we present the ontology-based interoperation service (OBIS), which calculates the matching of security concepts extracted from access requests and local authorization policies. This service is then validated in an employability use case scenario.

Keywords—authentication; authorization; security policies; ontology; ontology-based data matching; employability

I. INTRODUCTION

The Trusted Architecture for Securely Shared Services (TAS3) Integrated Project provides a next generation trust and security architecture that is ready to (1) meet the requirements of complex and highly versatile business processes, (2) enable the dynamic user-centric management of policies and (3) ensure end-to-end secure transmission of personal information and user-controlled attributes between heterogeneous, context dependent and continuously changing systems. This includes a trust and data protection infrastructure for managing and assessing the risks associated with identity authentication and the trustworthiness of actors in a federated infrastructure.

One of the main challenges, in this context, is to guarantee the correct interpretation and implementation of data protection policies, while sharing, accessing, and using information processing services in federated environments. These are essential elements of trust that are required for various stakeholders, especially end-user, to participate in any implementation of the TAS3 architecture. As new service providers join a TAS3 federation, with new policies that are protecting their resources and new credentials that are assigned to their users, it cannot be guaranteed that all federation members will use the same terminology for the same concepts. To solve this, some federation have mandated that all members use the same vocabularies, credentials and policy terms. But this is unrealistic in large scale multi-national federations.

The TAS3 approach is, instead, to introduce an ontology-based interoperation web service (OBIS). This is configured with the vocabularies of all the service providers, and consequently is able to calculate the dominance relationship between two security concepts inferred from an access request term and a local authorization policy term, based on the generic ontology-based data matching framework (ODMF) [1]. The OBIS web service differs from existing approaches by providing a method for the association of policies and controls from all the service providers in a federated system and being integrated into the authorization infrastructure. In the proposed approach, every stakeholder expresses his authorization policy using his own vocabulary, and when a policy engine receives an authorization request containing an unknown term, it semantically matches this to one that is locally known by the authorization policy. This linguistic-based approach is adopted in the privacy domain in order to allow different non-technical users (and organizations) to express their security policies in an intelligible way, through the use of natural language, thus enforcing the user-centricity aspect.

The rest of the paper is organized as follows: section II briefly describes related work, whilst section III provides background information on the technology being used. The approach is presented in section IV. Section V validates the approach on an employability use case scenario developed by the University of Nottingham [2]. Section VI presents our conclusion and suggestions for future work.

II. RELATED WORK

Several approaches exist which aim at resolving semantic access controls.

The Semantic Access Control (SAC) Model [3] was specifically designed to enforce ABAC policies in heterogeneous and distributed environments. It maps policies to resources dynamically based on the semantics of policies and resources. The Semantic Access Control Enabler (SACE) [4] was developed to enforce Role-Based Access Control (RBAC) when accessing databases from heterogeneous databases.
Verma [5] presents a semantic policy matchmaking for web service policies specified across multiple domains (e.g. security, privacy, trust).

Kaos [6] is a semantic policy language and a framework for the specification, management and enforcement of policies within different security domains. A similar approach is presented in [7], concerned with the meaning of contexts to be used directly in an access control policy.

III. BACKGROUND

The ontology described in this paper adopts the paradigm of Developing Ontology Grounded Methodology and Applications (DOGMA, [8]). In DOGMA, an ontology consists of a set of lexons and a set of commitments.

A lexon is defined as a quintuple \( \langle \gamma, t_1, r_1, r_2, t_2 \rangle \) representing a fact type. \( \gamma \) is a context identifier that points to a context where two terms, \( t_1, t_2 \) are originally defined and disambiguated. \( r_1, r_2 \) are two roles that characterize the relationship between \( t_1 \) and \( t_2 \). For example, \( \langle \text{ABAC, subject, is assigned, is assigned to, attribute} \rangle \) is a lexon which means “in the context of ABAC, a subject is assigned an attribute and an attribute is assigned to a subject”. TABLE I illustrates lexons representing the high level concepts of an ABAC policy, extracted from the security policy ontology (SecPODE, [9]).

<table>
<thead>
<tr>
<th>Head term</th>
<th>Role</th>
<th>Co-role</th>
<th>Tail term</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecurityPolicy</td>
<td>controls</td>
<td>controlled by</td>
<td>Action</td>
</tr>
<tr>
<td>SecurityPolicy</td>
<td>has</td>
<td>of</td>
<td>Target</td>
</tr>
<tr>
<td>SecurityPolicy</td>
<td>written by</td>
<td>writes</td>
<td>Subject</td>
</tr>
<tr>
<td>Action</td>
<td>performed by</td>
<td>performs</td>
<td>Subject</td>
</tr>
<tr>
<td>Action</td>
<td>performed on</td>
<td>under</td>
<td>Target</td>
</tr>
<tr>
<td>Target</td>
<td>subsumes</td>
<td>is a</td>
<td>Resource</td>
</tr>
</tbody>
</table>

A commitment contains a constraint on a (set of) lexon(s). For instance, we can apply the mandatory constraint on the above lexon, \( \langle \text{there exists at least one attribute assigned to each subject} \rangle \). The commitment language needs to be specified in a language such as OWL (http://www.w3.org/TR/owl-ref/) or SDRule language [10].

Several access control paradigms have been proposed over the years. In our approach, we use an enhancement of the ISO Attribute Based Access Control (ABAC) model [11].

IV. APPROACH

The role of the ontology-based interoperation service is to provide semantic interoperability between service requesters and service providers based on the security policy ontology (SecPODE). The main assumptions are that (1) inputs are fully qualified URIs and (2) the service requesters and the service providers both commit to the SecPODE ontology. OBIS then performs the matching between two concepts expressed in different security policies, focusing for now on three main categories: subjects, actions and targets (see TABLE I). More specifically, OBIS returns a value representing the relation between terms within the same category.

A. Ontology-based Security Attributes Matching

The security concepts level of domination is calculated using the generic ontology-based data matching framework (ODMF).

Given an ontology, ODMF (1) maps data into semantic networks (Tree, Directed Acyclic Graph / lattice or any directed graphs); (2) performs semantic computation, such as path recognition (shortest path, connectivity), path strength in scores (semantic vicinity, etc.), composite semantic similarity of semantic networks; (3) performs literal computation, such as fuzzy similarity of literals (strings); and (4) performs lexical computation, such as synonymous similarity and similarity based on a user dictionary.

Each ontology-based data matching strategy in ODMF contains at least one graph algorithm. In this paper, we focus on the Controlled Fully Automated Ontology Assisted Matching (C-FOAM) strategy. We apply C-FOAM to compare two security terms annotated with security policy concepts.

C-FOAM contains two modules (see Figure 1) – the interpreter and the comparator – corresponding to the two components of OBIS. The interpreter module makes use of the lexical dictionary and the domain ontology to interpret the input term. Given a term that denotes a concept in the domain ontology, the interpreter will return the correct concept defined in the ontology. The comparator then uses any combination of the different graph algorithms for the path recognition between the two concepts (the policy concept and the access request concept).

In this paper, due to the specificity of the security terms (i.e. the domination relation), we only use semantic relations which grow or shrink a set (e.g. ‘is-a’ or ‘part-of’). C-FOAM uses AND/OR graphs for computing the dominance relation between the two concepts identified by the interpreter.

B. Ontology-based Interoperation Service

OBIS was designed as a web service providing an interface to perform relation lookups between two terms originating from security policies. The main method of the web service interface is summarized in Figure 2.
where URI(SR), URI(SP) are the two input values, representing the URIs of the two terms to be matched, and Val represents the returned value, i.e. the level of dominance between URI(SR) and URI(SP). There are six possible return values, as follows:

-3: “I don’t know SP”
-2: “I don’t know SR”
-1: “SR is not related to SP”
0: “SR is less general than (is dominated by) SP”
1: “SR is equivalent to SP”
2: “SR is more general (dominates) SP”.

The OBIS architecture, illustrated in Figure 3, contains two important components (the translator and the path finder).

The translator takes an input string corresponding to a term in the user’s/system’s terminology and maps it into a concept from the ontology. As a first step, we implemented a many-to-one mapping (i.e. many terms in the user’s terminology map to one concept in the ontology). The translator uses a terminology base (data repository), where each service provider has to contribute his own vocabulary (terminology) and say how this is mapped to the security policy ontology. We assume that this is performed by the security officer of each service provider, and that he can be trusted to perform an accurate mapping.

The path finder takes the output of the translator (i.e. two concepts, C1 and C2 from the ontology, corresponding to the two input strings) and returns the path (level of dominance) between them. The path is searched for in the Directed Acyclic Graph (DAG) corresponding to the Subject, Action or Target category of the ontology. The path finder is exemplified in TABLE II, where pairs of concepts are compared from the different DAGs (actions, subjects and targets). The level of dominance is calculated based on ontology-based data matching strategies, introduced in the previous section. A more detailed description is given in section V, TABLE III.

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>Level of dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Delegate</td>
<td>0 (less general)</td>
</tr>
<tr>
<td>University staff</td>
<td>Employer</td>
<td>-1 (not related)</td>
</tr>
<tr>
<td>ePortfolio</td>
<td>Education</td>
<td>2 (more general)</td>
</tr>
</tbody>
</table>

C. Authorization Architecture

The OBIS service is located in the authorization infrastructure as shown in Figure 4. The authorization process consists of the following steps. The Service Requester (SR) launches a request to access a resource protected by various authorization policies. The resource’s Policy Enforcement Point (PEP) intercepts the request and passes it to the Application Independent Policy Enforcement Point (AIPEP). This initially contacts the Credential Validation Service (CVS) to validate the requestor’s credentials (i.e. attribute claims) which typically have been issued by multiple different attribute authorities in the federation (not shown in Figure 4). The CVS uses its local credential validation policy to determine which credentials are valid. If the CVS cannot validate any credential, it contacts OBIS to determine the relationship between the presented attributes and the ones in its policy. Based on the domination relationship returned by OBIS, the CVS is able to return the set of valid attributes to the AIPEP. The AIPEP now contacts the Master PDP, passing the valid attributes, and asking for an access control decision. The Master PDP calls the set of subordinate PDPs which support different access control policy languages (e.g. XACML [12][13] and PERMIS [14]). This ensures that all service providers do not need to support the same policy language, and that policies in different languages can be passed between providers and still enforced by them. If a PDP is not able to make an access control decision, then it calls OBIS to calculate the semantic relation between terms in the access request. Based on the values returned by OBIS, the access request is either granted or denied.

V. USE CASE SCENARIO

Within the TAS³ project, we have developed an employability demonstrator focusing on the management of internships and work placements for university students [2]. Timely, accurate and secure presentation and exchange of verified skills data and personal information is key to the success of this scenario. For example, recruiters and prospective employers want to access verified data in a...
standardized format (e.g. HR-XML\(^1\)) to facilitate comparisons and matching of students with job profiles. Similarly, candidates want to retain control over how their personal information is accessed, processed and stored by third parties by setting their own security policies.

![Diagram](image)

Figure 5. The employability scenario.

Figure 5 represents one of the employability scenarios in TAS\(^2\). In this scenario, Alice is a second year student at a UK university and seeks a summer work placement. Alice contacts a placement service approved by the university to discuss the details of her application. Her placement advisor, called Bob, informs her that he first needs to verify that she is a registered student at the university. Once Bob has received the confirmation, he contacts Alice to get permission to access relevant information to match her to available placements. Alice is happy to share this information subject to this information not being shared to third parties without her approval. Based on this information, Bob identifies a number of placement providers that he believes have suitable placements for students like Alice. Alice wishes to be put forward for two placements and agrees that the placement advisor can act on her behalf to agree terms of a work placement and she consents to have relevant personal information to be disclosed to them. Bob forwards Alice’s information to the placement providers for consideration.

In this scenario, several security and trust issues may be encountered, as follows:

- Does the student trust that the placement advisor is approved by the university?
- Can the student trust that only relevant personal information is used during the placement process?
- Can the student trust that the information provided to the placement provider is protected as per her privacy policy?

The placement advisor, placement providers, etc. use their own systems to store the information. How can all stakeholders be sure that personal information is secure between one placement and another?

When Alice issues a request for a placement or internship, the placement service first verifies with the Identity Provider (IdP) that the person is who she claims to be (see Figure 6).

![Diagram](image)

Figure 6. Architecture to enforce security and privacy policies.

After this, Bob needs to access relevant information about Alice from distributed data repositories (e.g. a CV from within the student’s ePortfolio and university repositories). The information stored in these repositories is protected by Alice’s privacy policy as well as the keeper’s own policy. If Alice’s privacy policy states that only members of the university have permission to access her work, then Bob’s request to access a subset of her work needs to be validated against this policy to see whether he can gain access or not. OBIS is needed to aid this access control decision.

The role of OBIS in the authorization architecture is to determine whether (1) a foreign subject dominates the subject in the authorization policy, (2) the requested action is dominated by the action in the access control policy and (3) the resource to be accessed by the subject is dominated by the resource in the policy. In addition OBIS may be called directly by the placement application to determine if information in Alice’s documents fulfil the requirements of the various placement providers. Some examples from the employability scenario are illustrated in TABLE III.

<table>
<thead>
<tr>
<th>OBIS Role</th>
<th>Security Concept Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>To check whether Bob can access Alice’s personal data by calculating the relation between the placement advisor (Bob) role in the request and the University Staff role in Alice’s policy (to determine whether a foreign role dominates the local role in the credential validation policy).</td>
<td>Role matching for CVS.</td>
</tr>
<tr>
<td>Translator: SR: Bob = delegate_right</td>
<td>SP: University Staff = delegate_right</td>
</tr>
<tr>
<td>Path finder: 1 “equivalence” Attribute is valid</td>
<td></td>
</tr>
<tr>
<td>To determine whether the resource to be accessed by the service requester (e.g. Alice’s previous employment experience) is more specific than the resource protected by the security policy (e.g. Alice’s CV).</td>
<td>Resource matching for PDP.</td>
</tr>
<tr>
<td>Translator: SR: Alice previous employment experience = work experience</td>
<td>SP: Alice’s CV = CV</td>
</tr>
<tr>
<td>Path finder: 0 “less general” Access granted</td>
<td></td>
</tr>
<tr>
<td>To determine whether Alice’s programming competency (resource in the CV) is more specific than the requested competency (“the candidate should have good programming skills”) for a vacant job.</td>
<td>Resource matching for application.</td>
</tr>
<tr>
<td>Translator: SR: Java programming = competencies</td>
<td>SP: the candidate should have</td>
</tr>
</tbody>
</table>
The implementation of a disambiguator, whose purpose is to determine the semantic relationships between additional request parameter terms and those in the authorization policies. This implies separation of duties. Once the ontology exists, security administrators will be able to map their additional vocabularies to the security policy ontology underlying the OBIS service, without needing to know specific details about the security domains of remote service providers; (3) automated, through an integrated architecture which ensures OBIS is called by credential validation services in a web service to operate in an open, distributed and dynamic environment; and (5) secure, enabling query-only requests via SSL/TLS links.

Future work will involve developing the ontology for more sophisticated authorization policies, including such concepts as obligations, delegation of authority, environmental parameters and separation of duties. Once the ontology exists, security administrators will be able to map their additional vocabularies into this, enabling the policy engines to call OBIS in order to determine the semantic relationships between additional request parameter terms and those in the authorization policies.

We envisage, as future work, a many-to-many mapping for the translator component, which is more realistic. This implies the implementation of a disambiguator, whose purpose is to decide upon the correct correspondence between the user vocabulary and the ontological concepts.

Our next step is to validate the OBIS service against an eHealth application within the TAS\textsuperscript{3} project and also against other components of the TAS\textsuperscript{3} architecture (e.g. using OBIS for semantic interoperability in creating secured business processes).

**ACKNOWLEDGMENT**

This paper is supported by the EC FP7 TAS\textsuperscript{3} (Trusted Architecture for Securely Shared Services) project. The authors would like to thank all TAS\textsuperscript{3} project partners for their contribution to the research.

**REFERENCES**


