TOWARDS AN AUTHORIZATION SYSTEM FOR CLOUD INFRASTRUCTURE PROVIDERS

Jorge Bernal Bernabe*, Juan M. Marin Perez*, Jose M. Alcaraz Calero$,
Felix J. Garcia Clemente+, Gregorio Martinez Perez* and Antonio F. Gomez Skarmeta*
*Departamento de Ingenieria de la Informacion y las Comunicaciones, University of Murcia, Murcia, Spain
$Cloud and Security Lab, Hewlett-Packard Laboratories, Bristol, U.K.
+Departamento de Ingenieria y Tecnologia de Computadores, University of Murcia, Murcia, Spain

Keywords: Authorization system, Cloud computing, Multi-tenancy, Trust model, Semantic web.

Abstract: The provision of security services is a key enabler in cloud computing architectures. Focusing on multi-tenancy authorization systems, the provision of different models including role based access control (RBAC), hierarchical RBAC (hRBAC), conditional RBAC (cRBAC) and hierarchical objects (HO) is the main objective of this paper. Our proposal is based on the Common Information Model (CIM) and Semantic Web technologies, which have been demonstrated as valid tools for describing authorization models. As the same language is being used for the information and the authorization models they are both well aligned and thus reducing the potential mismatch that may appear between the semantics of both models. A trust model enabling the establishment of coalitions and federations across tenants is also an objective being covered as part of the research being presented in this paper.

1 INTRODUCTION

Nowadays, many businesses are evolving to start using Cloud Computing (Hayes, 2008) architectures as a new way of managing data centers, enabling efficient provisioning of virtual IT architectures. Such virtual resources are dynamically created and dismantled on-demand, providing new pay-as-you-go business models for the usage of infrastructures.

To achieve an effective management of the data centers, cloud computing has provided a well-known logical stack. This stack is divided in three different layers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). From the point of view of the cloud provider, the access control model in the IaaS layer becomes a critical aspect to be considered in order to have a fine control over the usage of the architecture.

Current cloud vendor providers such as Amazon EC2 [1], GoGrid [2] and Rackspace [3] rely on mere authentication schemes which do not provide access control services to the resources managed beyond total access as administrator to the whole system. During the last few months, some authorization systems have appeared trying to solve this issue. However, current solutions lack of enough expressiveness to describe some real authorization business policies, which can be a differentiating key for the selection of one cloud provider or another.

The main aim of this paper is to describe an authorization system suitable for cloud computing architectures which solve some of these lacks, providing a multi tenancy authorization system with high level of expressiveness in the definition of grants. Moreover, the establishment of business alliances (coalitions and federations) by means of a trust model is also considered as part of the proposal.

This paper has been structured as follows: section 2 provides a related work overview. Section 3 provides an introduction of the languages used in this approach. Section 4 describes the authorization model. Section 5 explains the architecture and the process carried out for authorization. Finally, section 6 provides some conclusions and states of direction.

2 RELATED WORK

Some novel authorization models for cloud computing have recently appeared during the last year. Thus, (Alcaraz-Calero et al., 2010a) has provided

[1] Amazon EC2 available at http://aws.amazon.com/es/ec2/

[2] GoGrid available at http://www.gogrid.com/

[3] Rackspace available at http://www.rackspace.com/index.php

Bernal Bernabe J., M. Marin Perez J., M. Alcaraz Calero J., J. Garcia Clemente F., Martinez Perez G. and F. Gomez Skarmeta A..
TOWARDS AN AUTHORIZATION SYSTEM FOR CLOUD INFRASTRUCTURE PROVIDERS.
DOI: 10.5220/0003525703330338
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2011), pages 333-338
ISBN: 978-989-8425-71-3

an advanced multi-tenancy authorization model with RBAC support based on authorization statements defined by means of paths. (Danwei et al., 2009) has provided another authorization system based on *usage control* access model (UCON) (Park and Sandhu, 2004) and negotiation technologies.

While the previous authorization systems are suitable for cloud computing, they may lack of enough expressiveness for defining high level authorization policies. This lack of expressiveness can be addressed under the usage of the Semantic Web technologies and ontological approaches since they provide a significant level of expressiveness. Some authorization systems have been proposed under Semantic Web. Thus, (Alcaraz-Calero et al., 2010b) provides a complete multi-tenancy authorization model for distributed system with support for RBAC, hRBAC, cRBAC and authorization policies. (Perez et al., 2011) proposed other semantic-aware multi-tenancy authorization system for grid architectures.These previous proposals are not directly designed for cloud computing but they may be considered as the basis for the designing of a semantic-aware authorization model for cloud computing.

Regarding semantic-aware proposals designed for cloud computing, (Hu et al., 2009) analyses existing access control methods and present a new Semantic Access Control Policy Language (SACPL) for describing access control policies in cloud computing environment. Although, this is a prominent research work, authors only provide simple RBAC support and they do not describe the semantics of the authorization model proposed hampering the evaluation of the quality of such approach.

3 INFORMATION MODEL FOR AUTHORIZATION

A model is used to represent authorization related concepts as well as the resources to be protected by the authorization system. This model is represented by means of ontologies, since they provide high expressiveness to define the different concepts and their semantics. The ontologies being represented in this model have been defined using the Ontology Web Language 2 (OWL 2). This is a W3C standard which enables the specification of ontologies. The Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) is used to represent rules on the Semantic Web and it extends OWL 2 in order to provide a way to express conditional knowledge. These languages have a common basic logic formalism called Description Logic. This formalism provides a highly expressive language for describing semantic features such as inheritance among concepts and properties, transitiveness and reflexiveness in properties, disjointness of concepts and conditional knowledge.

In order to represent the resources to be protected by the authorization system, we chose a reference model to generate the ontology which models the concepts and semantics of the underlying managed virtual infrastructure. The Common Information Model (CIM) (Bumpus et al., 2000) created by the DMTF has been selected as base model for our authorization system. Several reasons have motivated this choice: it is a quite complete information model, it supports extensibility mechanisms and it is an standard information model. Moreover, there are related standards and technologies grouped under the WBEM specifications which allow to dynamically gather the current state of the underlying infrastructure by means of CIM. Additionally, this information model has been used in several research works like (Debusmann and Keller, 2003), (Mao et al., 2006), among others, and it is used in a wide variety of large systems such as SAP, Microsoft Windows and VMWare, among others.

CIM provides many concepts and relationships to represent an information system. However, the model used in our authorization system is intended to represent only the managed resources and not all the information is really needed. Thus, a subset of CIM is used. The concrete subset depends on the elements and capabilities of the virtual infrastructure to be managed. Common concepts managed in cloud infrastructure include virtual machines (VMs), virtual networks which interconnect the VMs, volumes attached to the VMs, etc. The model should contain enough concepts and relationships to define the different elements to be protected by the authorization system. The model can also be extended to incorporate specific concepts making use of the extensibility mechanisms provided by CIM.

Different representations of CIM in OWL are provided in several works like (Majewska et al., 2007), (Heimbigner, 2004) and (Alcaraz-Calero et al., 2010b). In this paper, the approach proposed by (Alcaraz-Calero et al., 2010b) has been selected to perform the representation of CIM in OWL. This approach is available at the XCIM2OWL Sourceforge project.

4 AUTHORIZATION

Authorization decisions are taken based on some privileges defined for some subjects on some resources. In this approach, privileges are established by means of authorization statements. An authorization statement basically establishes that a subject has a privilege on a resource. It can be represented by the 3-tuple *(Subject, Privilege, Resource)*.

In order to support a Role-based Access Control (RBAC) model, a subject can be either a user or a role. These authorization statements can be represented in the model by means of the set of CIM concepts and relationships depicted in Figure 1.

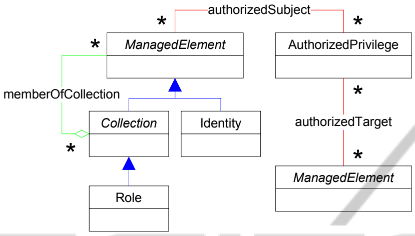

Figure 1: UML diagram of authorization concepts.

Subjects are represented by the `Identity` and `Role` concepts. The association of subjects to privileges is accomplished via the `authorizedSubject` association. The resources of the underlying virtual infrastructure that are protected are similarly defined via the `authorizedTarget` association. This association connects with the `ManagedElement` concept, which is the common root superclass in CIM. This enables the definition of any element of the model as target for authorization.

These elements of the model represent a positive authorization statement. When a user tries to perform some action on a given resource in the system, the authorization system searches the model for the set of concepts and relationships. If no positive authorization statement is found in the model, then the access to the resource will be denied, i.e. a 'deny by default' policy is followed.

In this approach, SWRL is used to define rules based on the concepts represented in the model. In particular, two different kinds of rules are considered: rules and meta-rules. Rules are used to map the authorization statements represented by the tuples to the corresponding concepts and relationships of the model which represent the specified semantics of the statement. In turn, meta-rules are rules which are defined to extend the semantics of the model, thus enabling support for different authorization features.

One of these features is Role-based Access Control (RBAC) support, which enables the definition of privileges for roles. A meta-rule is defined to provide the RBAC semantics by propagating the privileges assigned to a role to the identities belonging to that role. Rule 1 shows this meta-rule in SWRL abstract syntax.

$$Role(?r) \wedge AuthorizedPrivilege(?p) \wedge authorizedSubject(?p,?r) \wedge \quad (1)$$

$$Identity(?i) \wedge memberOfCollection(?i,?r) \wedge \quad (2)$$

$$\rightarrow \quad (3)$$

$$authorizedSubject(?p,?i) \quad (4)$$

Rule 1: Role privileges propagation meta-rule.

The first line of Rule 1 defines a role which has some privilege associated. Line 2 selects the identities which are members of that role and the rule consequent establishes the `authorizedSubject` association to grant the privilege to these identities.

Hierarchical Role-based Access Control (hRBAC) extends RBAC with the ability to define role hierarchies. In our approach, these hierarchies can be defined by means of the `memberOfCollection` association. Privilege inheritance is achieved by establishing the semantics of this hierarchical relationship in the model. The definition in OWL of the `memberOfCollection` property as transitive makes the system to consider any instance belonging to any role, also belonging to its parent roles in the hierarchy. Then, for any given instance, rule 1 will propagate the privileges in the role hierarchy.

Enabling object hierarchies in the model provides a higher expressiveness to the authorization system. This feature enables the definition of privileges affecting a given object to be inherited by its children objects in the hierarchy. For example, privileges affecting a directory would also apply to its subdirectories and files. This approach can also be applied to compound objects (e.g. a privilege defined over a network may also affect to each element which forms part of such a network). This feature is achieved by incorporating the corresponding semantics to the model and by defining meta-rules which establish the privilege propagation. In CIM, associations used to establish 'part of ' and dependency relationships share the common root `Component` and `Dependency` associations, respectively. The definition of these two root associations as transitive in OWL enables the authorization system to recognize all subcomponents and dependent objects through the whole hierarchy. The SWRL meta-rules which propagate the privileges are defined in a similar way as rule 1.

Conditional RBAC is also supported in this approach. It is done by enabling the granting of privileges according to some conditions. In order to provide this feature, the authorization statements are now represented by a 4-tuple *(Subject, Privilege, Resource, Conditions)*. This means that the permissions defined in the statement only apply when the specified conditions are fulfilled. The expressiveness provided by SWRL has been used to provide this feature, including some antecedents to the rule which maps the statement to the model. Rule antecedents are generated from the conditions specified by the administrator in the 4-tuple. Authorization statements which does not specify any condition will result in a rule with an empty antecedent.

Finally, multi tenancy support is also needed in this environment, since usually multiple entities share the same cloud provider and it has to ensure a secure access to the information. This feature is provided in our authorization system by keeping the authorization models of the different administrative domains isolated at the level of knowledge. Supporting this multi tenancy feature, also implies a new extension to the authorization statements, adding the concept of Issuer, which represents the entity which defines the statement. Thus, the authorization statements are now represented by the 5-tuple *(Issuer, Subject, Privilege, Resource, Conditions)*. This tuple can be read as *’The entity* Issuer *states that the user or role* Subject *has the permissions defined by* Privilege *over the element* Resource*’*.

Related to the multi tenancy feature, coalition support is also provided in our approach by enabling different organizations to collaborate in the Cloud. Trust relationships among different domains can be defined, meaning that an entity is able to access the knowledge of another entity if a trust relationship is established. The management of these different entities knowledge will be further explained in section 5.

5 ARCHITECTURE

This section describes the cloud architecture designed for the authorization model described in sections 3 and 4. As can be seen in figure 2, the cloud computing defines a stack composed of three layers: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (Saas). Infrastructure as a Service is the delivery of computer hardware, i.e. servers, networking technology, storage, and data center space, as a service. The layer also includes the delivery of on-deman virtualization infrastructures to manage the resources. The functionality of managing these virtual infrastructures is provided by means of an `IaaS API`.

Our semantic-aware authorization architecture, which is depicted in Figure 2, has been designed to be suitable for the IaaS layer. The architecture, following a multi-tenancy approach, provides access control to the infrastructure resources managed by the IaaS according to the authorization rules defined previously by customers. Our solution extends the already existing IaaS API solutions in order to cope with a fine grain authorization mechanism based on a high level of expressiveness. Thus, when a customer invokes the `IaaS API` trying to access to a resource of the virtual infrastructure, the `Interceptor` module captures the request and delivers it to the `SemanticAuthzService` in order to evaluate whether the customer has grants to access to the resource or not. After checking the access control, the `Interceptor`, in case of an affirmative response from the `SemanticAuthzService`, forwards the request again to the IaaS API which performs the corresponding action.

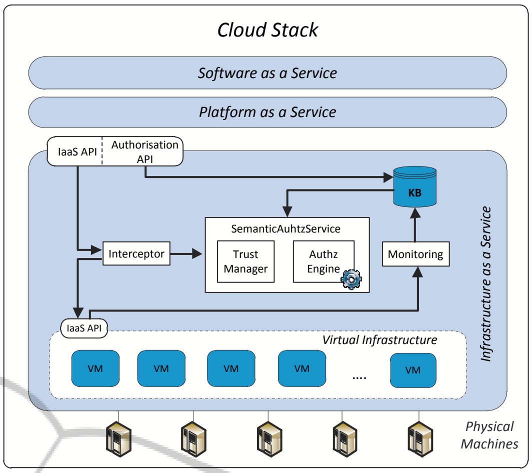

Figure 2: Semantic Authorization Architecture for Cloud.

As explained in section 4, the authorization rules required to perform the authorization decision are obtained from the authorization statements. The statements are configured and inserted in the system by users (either customers and/or system administrators) making use of the `Authorization API`. This API, taking these statements definitions as input, translates them as authorization rules in SWRL format in order to be later stored in a central repository called `Knowledge Base(KB)`. The KB holds all the authorization information needed to carry out the authorization decision. It includes, the authorization rules and the information model defining the virtual infrastructure being managed.

Since the `SemanticAuthzService` requires information about the underline virtual environment, the information stored at the `Knowledge Base` has to be kept up to date with the information managed by the IaaS. The `Monitoring` module is the authorization system component in charge of obtaining this kind of information from the IaaS API and translating it to the CIM-based ontology model used in the KB, as described in section 3.

The authorization service is composed of two main components the Authorization Engine and the Trust Manager. The former can be seen as the core of the authorization system since it takes authorization decisions reasoning over the SWRL rules and the ontology model. The latter manages the privacy of the information of the different organizations which make use of the cloud infrastructure according to the trust relationships existing between them.

5.1 Trust Management

A collaboration agreement between two organizations can be represented by means of a trust relationship (A trusts B). This means that a user of the organization B can access to the authorization rules as well as to the managed resources from organization A. This enables users of the organization B to define his own authorizations rules taking into account the particular domain information and the authorization model of A.

Thus, if there is a trust relationship between A and B, the Authorization Engine can use the authorization rules as well as the ontology model hold in the KB defined for A when a authorization decision has to be taken for an organization B. This allows organizations to control the access to their authorization information and the resources of the IaaS layer they use. Keeping the information model and the authorization rules private implies that nobody outside of the established agreement will be able to define or use rules allowing members of untrusted organizations to access to private resources. By default nobody trusts anyone else unless there is an explicit statement of this. Only a user of a given domain can insert or remove rules regarding her domain. As a result, the Trust Manager controls the privacy of all the information stored in the Knowledge Base.

The TrustManager is in charge of selecting the information to be used to make the authorization decision. This choice is done taking into account the Issuer to provide multi-tenancy in the cloud environment. There is one KB for each organization containing its own domain model and rules as well as the knowledge of its trusted organizations. The TrustManager is also responsible of managing the life cycle of the different KBs according to the trust relationships between the organizations.

When a user tries to access to a resource, the request reaches firstly to the IaaS API. The Interceptor takes the request and forwards it to the SemanticAuthzService. Then, the Trust Manager selects the corresponding KB according to the organization to which the request target belongs. Once the Authorization Engine knows about the appropriate KB where performing the authorization query it reasons over the model and rules deriving the authorization decision. Then, the authorization response is sent back to the Interceptor which forwards it to the IaaS-API that finally grantees or denies the access to the resource based on the decision.

5.2 Reasoning Process

The information about the underline virtual infrastructure as well as the authorization model and constraints are specified as an OWL ontology. AuthzEngine performs authorization decisions based on the reasoning process performed by an OWL and SWRL reasoner. During the reasoning process, the reasoner uses the knowledge base which contains the OWL ontology model together with the authorization rules and meta rules defined in form of SWRL rules. At these stage, two kinds of reasoning can be identified: OWL reasoning and SWRL reasoning. Whereas the former deal with operations of querying, validation and inference about the ontology, the latter consists on performing inference with SWRL rules. The inference process (about both the OWL ontology and the SWRL rules) generates new knowledge updating the KB. Validation is also performed, checking the ontology for inconsistencies or constraint violations.

Usually during the reasoning process, the SWRL authorization rules infers new *OWL AuthorizedPrivilege* instances which represent an authorization definition according to our CIM based model depicted in Figure 1. Thus, these instances are present in the KB as the result of the inference from a rule representing an authorization statement. To perform the decision, the AuthzEngine component queries the reasoner looking for these OWL instances in the KB. Listing 1 shows the authorization query which looks for the privilege allowing a subject to access to a given target when he tries to perform certain action.

```
PARAMS (subject, target ,  action )
SELECT(?p)
WHERE AuthorizedPrivilege(?p) ∧ action(?p, ,#action) ∧
Identity(?s) ∧ instanceID(?s, #subject) ∧
authorizedSubject(?p,?s) ∧ authorizedTarget(?p, #target)
```

Listing 1: Authorization query.

The query explores the KB looking for an AuthorizedPrivilege instance which is associated with the given subject and target. The query has three parameters: the subject or identity which is trying to access, the target representing the cloud resource which is being accessed, and the action performed against the target. If the query does not return any privilege instance, the AuthzEngine denies the subject accessing the target (deny by default).

6 CONCLUSIONS AND FUTURE WORK

A multi-tenancy semantic-aware authorization model based on CIM has been proposed for cloud computing scenarios. This authorization model enables high-level authorization policies while overcoming certain lacks of expresiveness of its predecessors and providing support for advanced authorization features such as RBAC, hRBAC, cRBAC, HO and authorization policies. Semantic Web tecnologies has been demonstrated as useful for describing authorization models. Moreover, the usage of the same language for expressing both information and authorization models avoid any mismatch between the semantics of the information model and the semantics of the authorization model, which in turn, is a a potential problem available in most of the current authorization proposals.

As a future work, the implementation of an authorization plug-in for the Eucalyptus open source cloud provider, in which the authorization model explained here can be inserted in production scenarios is a desised result. It is also expected for the coming months to perform intensive performance analysis of the authorization model proposed in order to stablish an analitical comparison of the trade-off between language expressiveness and system performance. Moreover, another expected work is the extension of the authorization model proposed in order to include conflict detection capabilities in cloud computing architectures.

ACKNOWLEDGEMENTS

This work has been partially funded by the project ”Secure Management of Information across multiple Stakeholders (SEMIRAMIS)” CIP-ICT PSP-2009-3 250453, within the EC Seven Framework Programme (FP7). Thanks also to the Funding Program for Research Groups of Excellence granted by the Seneca Foundation (04552/GERM/06). Authors also thank to the the Seneca Foundation for the post-doctoral grant 15714/PD/10 sponsoring Jose M. Alcaraz Calero. Finally, authors would like to specially thanks Nigel Edwards for his great contribution to the authorization field.

REFERENCES

Alcaraz-Calero, J. M., Edwards, N., Kirschnick, J., Wilcock, L., and Wray, M. (2010a). Towards a multi-tenancy authorization system for cloud services. *IEEE Security and Privacy*, 8(6):48–55.

Alcaraz-Calero, J. M., Perez, G. M., and Skarmeta, A. F. G. (2010b). Towards an authorization model for distributed systems based on the semantic web. *IET Information Security*, 4(4):411–421.

Bumpus, W., Sweitzer, J. W., Thompson, P., Westerinen, A., and Williams, R. C. (2000). *Common information model: implementing the object model for enterprise-management*. John Wiley & Sons, Inc.

Danwei, C., Xiuli, H., and Xunyi, R. (2009). Access control of cloud service based on ucon. *LNCS Cloud Computing*, 5931:559–564.

Debusmann, M. and Keller, A. (2003). SLA-driven management of distributed systems using the common information model. In *Proceeding of the 8th IFIP/IEEE International Symposium on Integrated Network Management*.

Hayes, B. (2008). Cloud computing. *Communications of the ACM*, 51(7):9–11.

Heimbigner, D. (2004). DMTF - CIM to OWL: A Case Study in Ontology Conversion. In *Conference on Software Engineering and Knowledge Engineering*.

Horrocks, I., Patel-Schneider, P. F., Boley, H., andB. Grosof, S. T., and Dean, M. (2004). SWRL: A Semantic Web Rule Language combining OWL and RuleML. Technical report, W3C.

Hu, L., Ying, S., Jia, X., and Zhao, K. (2009). Towards an approach of semantic access control for cloud computing. *LNCS Cloud Computing*, 5931:145–156.

Majewska, M., Kryza, B., and Kitowski, J. (2007). Translation of Common Information Model to Web Ontology Language. *LNCS Computational Science - ICCS 2007*, 4487:414–417.

Mao, H., Huang, L., and Li, M. (2006). Web resource monitoring based on common information model. In *IEEE Asia-Pacific Conference on Services Computing*.

Park, J. and Sandhu, R. (2004). The ucon abc usage control model. *ACM Transactions on Information and System Security*, 7:128–174.

Perez, J. M. M., Bernabe, J. B., Alcaraz-Calero, J. M., Clemente, F. J. G., Perez, G. M., and Skarmeta, A. F. G. (2011). Semantic-aware authorization architecture for grid security. *Future Generation Computer Systems*, 27:40–55.