Towards security and trust management policies on the Web

Theo Dimitrakos, Brian Matthews, Juan Bicarregui

CLRC Rutherford Appleton Laboratory, Oxfordshire, OX11 0QX, UK.
t.dimitrakos@rl.ac.uk,b.m.matthews@rl.ac.uk,j.c.bicarregui@rl.ac.uk

Abstract: The World Wide Web can be considered as the *universe of network-accessible information* (available through your computer, phone, television, or networked refrigerator...). Today, this universe benefits society by enabling new forms of human communication and offering new opportunities to share knowledge. However, the need to provide e-services over integrated inter-organisational and open networks makes the task of managing such systems very challenging, especially in terms of establishing the trust which is vital for effective transactions.

Recent work on *policy based management* of networks and distributed systems provides promising solutions to these problems while *the Resource Description Framework (RDF)* provides a common foundation that many communities are using. RDF is used to put data onto the Web in a form that can be processed by machines without prior arrangement, through the use of a common data model and machine-interpretable data schemas. By providing the means to encode policy specification in RDF, we expect to improve the effectiveness of policy based management while assessing the expressive power of RDF as a modelling solution for the Web.

Key words: Policy based management, Resource Description Framework, security and trust policy specification, WWW.

1. INTRODUCTION

The issue of trust in e-commerce is central for businesses as electronic services based on ubiquitous media (e.g. Internet, WWW, mobile phones) proliferate. The expansion of the once local information infrastructure to an enterprise-wide and, today, global one intensifies the demand for integrating heterogeneous inter-organisational networks and e-services, thus making the management of such systems more challenging than ever. To meet the requirements for adaptive architectures and personalised services on demand, computer scientists have devised several techniques for supporting adaptive e-services (e.g. active networks, mobile agents, resource discovery and brokerage mechanisms, etc.). In the near future, the emergence of the GRID infrastructure [1] will further intensify this demand[1] while introducing additional complexity by requiring more effective management of distributed (information and computational) resources. Such developments increase the security concerns about access control and make the trust management tasks increasingly demanding.

In [2], we suggested that the needs for flexibility and scalability are better addressed by separating the security and trust management framework from the purpose of the application, and we sketched a trust management framework related to a formal model of trust in e-services. In this paper we emphasise the explicit declaration and exchange of trust policies by participating resources using a suitable policy specification language. The nature of this task, where statements made about the properties of systems (i.e. policies and requirements from agents) are transmitted across the network, and automatic reasoning is used to determine action from those statements, is in line with proposals under the *Semantic Web* activity [5] of the World-Wide Web Consortium (W3C) [3]. Consequently, we propose stating the policies and trust statements in the Resource Description Framework (RDF) [4], and use the emerging tools of the Semantic Web to support the reasoning.

The rest of the paper is structured as follows. In section 2, we review the evolution of policy based management solutions for distributed systems placing emphasis on the emerging policy specification languages. In section 3, we explain why existing Web tools, such as the Extensible Mark-up Language (XML) come short on providing an effective framework for policy specification and exchange. We conclude suggesting that the emerging tools of the on-going Semantic Web activity of W3C appear to be able to address the shortcomings of existing Web tools and proposing extensions to the on-going work of this activity in order to support the specification and deployment of security policies.

[1] We foresee that the view of access to, and use of, knowledge, information and computation resources as tradable commodities, which will accompany the establishment of the GRID infrastructure, will be the driving force behind this intensification.

2. POLICY BASED MANAGEMENT

Policy based management of distributed systems promises to provide reliable and flexible solutions to the increasing complexity of managing security and trust in networks and open distributed systems. Furthermore, the recent trend of separating policy specification (modelling) from the implementation of a system allows for the policy to be modified in order to dynamically change the system management strategy and therefore to dynamically alter the system behaviour without changing the underlying system implementation.

The last 20 years have seen the emergence of very different approaches to specifying security and distributed system policy. Recently developed policy specification languages aim to bridge this gap by providing a common framework for specifying security and distributed systems management policies. In the following paragraphs we survey some of the most promising approaches to date in policy specification placing emphasis on frameworks which support security policy specification in addition to the core of policy-based distributed systems management.

In the 1980's and early 1990's, the security community developed a summary of models relating to the specification of access control policy (see [6] for a survey). These models have evolved into work on role based access control (RBAC) [7] and role based management, where a "role" can be seen as a group of related policies pertaining to a position in an organisation [8]. Some of this work has already resulted into implemented system architectures and core technologies that provide the infrastructure needed for implementing policy based management solutions, such as [9],[10].

The late 1990's have witnessed the emergence of separate tools for supporting policy-based management and security modelling. Network component manufacturers, the IETF and DMTF concentrate on the development of information models [11],[12] focusing on the management of Quality of Service (QoS) in networks. See [13],[14],[15] for indicative examples.

The establishment of policy-based management solutions that cater for security has been impeded by the lack of some common language, which can provide a unified approach to supporting the concepts of the policy models emerging from different research communities. Most emerging policy specification languages aim to fill this gap and they place emphasis on some of the following aspects of security:

– *Authentication*, which aims at the establishment of the identity of the user.
– *Access control*, which limits the activity of legitimate users who have been successfully authenticated. Access control policies can be divided to "obligatory", "discretionary" and "non discretionary" policies, depending on whether the administrators have the authority to specify security policies to be enforced by the access control system.

A typical example of policies for access control are *authorisation policies*, which are designed to protect target objects and are conceptually enforced by the target objects. In practice, authorisation policy enforcement has to be delegated to one or more *enforcement agents* that intercept actions and perform checks on whether the access is permitted. These can be further divided to *positive* and *negative* authorisation policies; the former permit an action whereas the latter forbid an action and are used for revocation of access rights.

Access control management provides rules and procedures for governing the choice in the behaviour of the system towards legitimate users who have been successfully authenticated. This involves catering for the delegation and propagation of authority and for management tasks that must be performed when certain events occur. In [16] access control management is achieved by means of the following types of policy.

– *Delegation policies,* allow the subject of an authorisation policy (grantors) to delegate some or all of their access rights to a new set of subjects (grantees). Effectively, when a grantor performs a delegation action, a new authorisation policy is created.
– *Refrain policies*, are similar to negative authorisation policies but their interpretation is subject based; they have to be enforced at the subject, e.g. by a *policy management agent* representing the subject entity, and apply to the actions that the subject invokes. Refrains are particularly useful for situations where one does not trust the targets to enforce a policy.
– *Obligation policies* are event-triggered policies that carry out management tasks on a set of target objects or on the subject itself. From this point of view, they differ from the other types in that they are not concerned with managing essentially access control tasks. Obligation are particularly useful for automating the reaction of a system when security violations occur or where resources need to be reconfigured to meet changing QoS requirements. Obligation policies have a subject based interpretation, in that the subject (or the *policy management agent* representing the subject entity) needs to interpret the policy statement and perform the actions on the target.

An important element of each policy is the set of conditions under which the policy is valid; they must be made explicit in the policy specification. The validity of a policy however, may depend on other policies existing or running in the system within the same scope or context. These conditions are usually impossible or impractical to specify as part of each policy, and therefore need to be specified as part of a group of policies. In [16] constraints are divided in two categories:

- *Basic policy constraints*, which limit the applicability of a basic policy and are expressed in terms of a predicate, which must evaluate to true for the policy to apply. Policy constraints can be considered as conjunctions of such predicates, and can be time, state or action constraints.
- *Meta-policies*, specify policies about the policies within a composite policy or some other scope, and are used to define application specific constraints.

Policy composition mechanisms are necessary defining composite policy types, which group and interrelate policies in order to support scalable policy specification for complex information systems. As an indicative example, see the declarative, object-oriented language described in [16] which covers a wide range of basic policies, enjoys policy composition and conflict resolution mechanisms, supports self-management by means of meta-policies, and allows the introduction of new types of policies as extensions or combinations of the existing basic policy types.

3. SECURITY POLICIES ON THE WEB

Traditionally, access policies are hard coded into the business logic of the implementation. With the emergence of flexible and adaptive *virtual organisations* this approach is no longer viable. Agents may seek to use resources from outside the organisation, and trust needs to be established with them. In general, these agents come from different computing architectures, thus have no knowledge of the security APIs provided by the requested resource. Further, as virtual organisations are extremely dynamic, with agents and resources constantly changing, trust policies need to be rapidly adapted; that is they need to be treated as a resource in their own right. For a more detailed discussion of the trust requirements in virtual organisations see [5]

As explained in [2] security and trust policies are particularly useful as a means of specifying trust intentions. The analysis and endorsement of trust intentions aims at the establishment of a dependable behaviour. This analysis is facilitated in the presence of a framework explaining how trust intentions are communicated between agents in distributed systems, therefore controlling the propagation of trust.

3.1 Security Policies on the Web: *our vision*

The *explicit* declaration and publication of security and trust policies by participating resources on the Web, can provide a basis for this framework. A suitable policy specification language such as the language described in [16], an object oriented approach to specifying access policies which covers a wide variety of access modalities, is required for describing policies. Policies can then be *encoded* in an appropriate resource description language, and *published* on the Web, as data objects associated with resources. Statements in this language have a standard machine interpretation that is universally accessible. Agents wishing to utilise resources would be able to present their credentials, own policies and requirements to the (agents representing) participating resources. An automated process would be able to verify the credentials, possibly referring to trusted third parties, establish identity, and deduce authorisation based upon the interpretation of the corresponding policies.

Agents representing resources would be able to automatically control access according to their interpretation of the (machine interpretable) rules that encode authorisation policies. Agents representing users will be able to *delegate* access to other agents according to delegation policies or will be asked to *refrain* from authorised actions in order to comply with a policy. In all these cases, agents representing users and resources will be able to publish policies, discover new policies, interpret new policies and incorporate them in their knowledge base thus expanding their policy meta-model.

Of course, publishing policy statements as resources on the Web makes them vulnerable to attack. Compromising the policy compromises the security of the whole site, and thus there is a high priority in protecting the integrity of the policy itself. However by treating a policy as a resource like any other, we allow for policies to be applied to policies. One can then use "hidden" policies for controlling access to "public" policies.

3.2 Security Policies on the Web: *what is still missing?*

The establishment of policy-based management solutions that cater for security has been impeded by the lack of some common language, which can provide a unified approach to supporting the concepts of the policy models emerging from different research communities. Such a language has to fulfil at least the following requirements:

- *Expressiveness*: The language should support policies to express management activity and security policies for access control, as well as delegation to cater for temporary transfer of access rights to agents acting on behalf of a client
- *Structure*: Sophisticated structuring mechanisms facilitate the specification of policies for large, complex systems and incorporate relating policies to collections of objects rather than individuals.
- *Compositionality*: Composite policies allow for basic security policies relating to roles, organisational units, and specific applications to be grouped. Composite policies become essential as the size and complexity of the targeted systems increases.
- *Conflict resolution*: The language should support the analysis of policy specifications for conflicts and inconsistencies. This assumes the ability to be able to determine which policies apply to an object and which objects a policy applies to. Declarative languages make this task easier.
- *Extensibility*: The language has to cater for new types of policies, which may arise in the future. An appropriate adaptation of the inheritance mechanism of the Object Oriented Modelling paradigm can support for achieving extensibility.
- *Comprehensiveness*: The language must be comprehensible and easy to use.

The Extensible Markup Language (XML) is the universal format for structured documents and data on the Web, and as such, it may as well be considered a candidate underlying linguistic framework on top of which one can build an appropriate policy specification language. Indeed, XML provides a means for designing textual descriptions of policies that are easy to generate and read (by a computer), that are unambiguous, and that avoid common pitfalls, such as lack of extensibility, lack of support for internationalisation/localisation, and platform-dependency. However, XML comes short on providing a *machine interpretation* of Web based data; XML DTDs and XML Schemas allow more flexibility in the syntactic descriptions of data, but do little to agree the interpretation of that data; each application domain has to agree on the meanings of terms. Providing a machine interpretation for Web based descriptions of security policy specifications is very important for avoiding security vulnerabilities associated with the misuse of a specified policy or its erroneous deployment. At present, a few security policy specification languages such as Ponder [16] provide a means of translating their policies to XML. Although, this is useful for viewing policy information with standard browsers, it is limited for publishing policies and sharing or exchanging policies between systems. There is no guarantee that XML descriptions of policies will be given consistent interpretations by systems that use a different policy meta-models. Even worse, new basic policies cannot be shared or exchanged even between systems with the same policy meta-model; basic policies have a semantic denotation that cannot be reduced to some combination or extension of other policy denotations.

3.3 Security Policies on the Web: *towards encoding security policies in RDF*

As mentioned in the previous section, the interpretation of documents and data on the Web has been largely determined by the user reading documents or establishing domain specific formats for XML data descriptions. There is no notion of machine readable descriptions of the interpretation of text on the web. Today as more sophisticated automated Web services are being demanded, the need for *machine interpretation* of web based data is increasing. In response to this demand, W3C has established a *Semantic Web* activity. The aim of this activity is to make the information presented on the web interpretable by machine. This activity is developing tools and techniques so that automatic agents can discover the relationship between machine resources, can interpret the meaning of those relationships and their attributes according to ontologies formally describing the properties of real world objects, and ultimately reason about the properties of these object in real-time to support services.

The basic mechanism underlying the Semantic Web activity is the *Resource Description Framework (RDF),* a language for describing relationships between objects. At its simplest level, RDF is a language for describing *triples* in the form of *(subject, predicate, object),* where the subject and the object of the triple are resources on the Web, and the predicate is some property of resources. More sophisticated statements can be made, including nested statements, bag and list, and reified statements, allowing the assertion of statements about statements. We shall introduce the necessary constructs in the course of the case study.

Particular classes of RDF models can be specified using *RDF Schemas.* This allows the user to specify *classes* and *subclasses* of resources, and *properties* and *sub-properties* between classes. However, RDF and RDF Schema are not sufficiently expressive to capture more than a simple graph model of objects and relationships. In order to realise the full potential of the semantic web, further components need to be constructed upon RDF.

- A means of defining more complex data models and ontologies with a more powerful constraint language.
- Logics to express rules and properties of RDF models and reasoning tools to derive true properties of RDF models.
- A language to express queries over set of RDF statements and have them evaluated in the logic.

The Semantic Web activity is planning to provide these technologies [5] and tools that implement them are likely to appear over time. Currently, there is research activity in all these areas. The most advanced is the DAML+OIL proposal [17] for capturing ontologies. We believe that as RDF matures, it can provide an effective means for specifying and exchanging security policies, as well as facilitate their deployment. However, it requires inventing machinery (orthogonal to RDF) addressing the encoding of security policies in RDF and the management of policy specifications as resources.

4. CONCLUSION

Our vision is to harness the Semantic Web activity to provide support for the deployment of security policies and thus facilitate the security and trust management based on standard tools built to support the Semantic Web. In this vision, policy specification statement could effectively be compiled into RDF rules, and the credentials of agents could then be presented as statements in (an extension of) RDF. Automated reasoning, supported by tools such as those mentioned in section 3.3, could then be used to establish identity, deduce authorisation based upon the interpretation of the corresponding policies. Agents representing resources would be able to automatically control access according to their interpretation of the RDF rules that encode authorisation policies. Agents representing resources and users will be able to *automatically control access and perform trust management tasks* according to the semantic interpretation of the RDF rules encoding policy specifications. They will also be able to publish policies, discover new policies, interpret new policies and incorporate their interpretation in their knowledge base thus expanding their policy meta-model.

The nature of RDF, which is designed for use with resources in a Web environment, is ideally suited when information (such as policy specifications and descriptions of deployment schemas) has to be shared, exchanged or retrieved from third parties to aid the trust establishment process. However, RDF and the semantic web tools are not yet developed. This makes it difficult to support, at present, the security policies as we require. Consequently we need to invent machinery to extend RDF to the required level[2]. This is a motivation for developing the Semantic Web further.

REFERENCES

[1] Foster I., and C. Kesselman (eds), *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Pub.,1998.

[2] Dimitrakos T., *System Models, e-Risks and e-Trust. Towards bridging the gap?* Towards the E-Society: E-Business, E-Commerce, and E-Government. (1st IFIP Conference on e-Commerce, e-Business, e-Government, Zurich, Switzerland), Kluwer Ac. Pub., 2001.

[3] *Semantic Web Activity Statement*, World Wide Web Consortium, 2001 http://www.w3.org/2001/sw/Activity

[4] Ora Lassila , & Ralph R. Swick (eds), *Resource Description Framework (RDF) Model and Syntax Specification*.W3C Recommendation 22 February 1999.

[5] Berners-Lee, T. *Semantic Web Road map*, W3C Note 1998 http://www.w3.org/DesignIssues/Semantic.html

[6] Clark D.D., and D.R. Wilson. *A Comparison of Commercial and Military Computer Security Policies.* In Proceedings of the IEEE Symposium on Security and Privacy.

[7] Shandu, R. S., E. J. Coyne, H. L. Feinstein, and C. E. Youman, *Role-Based Access Control Models*. IEEE Computer, 29(2): pp. 38-47, 1996.

[8] Lupu E.C., and M.S. Sloman, *Towards a Role Based Framework For Distributed Systems Management.* Journal of Network and Systems Management,. 5(1): pp. 5-30, 1997

[9] Sun Microsystems, Inc.. *Java Management Extensions Instrumentation and Agent Specification, v1.0,* December1999.

[10] Hegering, H.-G., S. Abeck, and B. Neumair. *Integrated Management of Network Systems*. Morgan Kaufman Publishers.

[11] DMTF, Inc. *Common Information Model (CIM) Specification*, version 2.2, June 1999. http://www.dmtf.org/spec/cims.html

[12] Moore, B., J.Strassner, and E. Ellesson, *Policy Core Information Model VI*. IETF Internet Draft, May 2000. http://www.ieft.org

[13] Goh, G., *Policy Management Requirements*. System Management Department, HP Laboratories Bristol, April, 1998.

[14] Hewlett-Packard Co, *A Primer on Policy Based Network Management*. Open View Network Management Division, HP, Sept. 1999.

[15] Internet Engineering Task Force, Policy Working Group. http://www.ietf.org/html.charters/management-charter.html

[16] Damianou, N., N. Dulay, E. Lupu, and M. Sloman, *The Ponder Policy Specification Language* .In Proc. Policy 2001: Workshop on Policies for Distributed Systems and Networks, Bristol, UK, 29-31 Jan. 2001, Springer-Verlag LNCS 1995, pp. 18-39

[17] DAML+OIL proposal, March 2001 http://www.daml.org/2001/03/daml+oil-index

[18] Matthews B., Bicarregui J., and Dimitrakos T., Building Trust on the GRID (Trust Issues Underpinning Scalable Virtual Organisations). CLRC Rutherford Appleton Laboratory, 2001.

[2] We note that the P3P initiative being pursued by the W3C would also have a need for expressing policy statements if it were to have automatic policy checking, as proposed by that working group.