Semantic Web Policy Systems

Matthew Dunlop
Using Semantic Web Technologies for Policy Management on the Web

Lakana Kagel, Tim Berners-Lee, and Dan Connolly
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Lab
Cambridge, MA 02139
{lkagel, timb, dconnoll}@ai.mit.edu

Abstract

With policy management becoming popular as a means of providing robust Web security, the variety of policy languages being proposed for the Web is constantly increasing. We recognize the importance of policies for securing the Web and believe that the future will only bring more policy languages. To make this a reality, however, there needs to be a way to bring them together in a single framework. In this paper, we present a methodology for using Semantic Web technologies to develop a set of general-purpose templates for building more detailed, domain-specific policy languages. Using these templates, we can create a policy language that can be applied to a single default policy domain. Instead there should be a way to incorporate domain-specific policies into this default framework, allowing for a single, consistent policy language that can be applied to all domains. This will help prevent the proliferation of independent policy languages and provide an engine for reasoning over the descriptions, both of which can be used to develop domain-specific policy languages.

Introduction

The Web is one of the most important ways for disseminating information across global boundaries. Though it is a simple and convenient framework for searching and retrieving information, the Web suffers from the lack of easy-to-use and adaptable security managed by website administrators, application developers, and people of all web content. Several approaches for access control in Web resources have been proposed such as WS-Policy (2005), TrustWeb (Gleave, et al. 2000), and SPKI/SPKI (Goldberg, 2002). However, the current approaches do not work well in all environments. This leads to the following problems:

1. Limited interoperability among frameworks that use different languages and policies.
2. Limited interoperability among frameworks that use different languages and policies.
3. Limited interoperability among frameworks that use different languages and policies.
4. Limited interoperability among frameworks that use different languages and policies.
5. Limited interoperability among frameworks that use different languages and policies.
6. Limited interoperability among frameworks that use different languages and policies.
7. Limited interoperability among frameworks that use different languages and policies.
8. Limited interoperability among frameworks that use different languages and policies.
9. Limited interoperability among frameworks that use different languages and policies.
10. Limited interoperability among frameworks that use different languages and policies.

An important aspect of secure policy management is the representation of policy languages. The current approaches use different languages and policies, which leads to the following problems:

1. Limited interoperability among frameworks that use different languages and policies.
2. Limited interoperability among frameworks that use different languages and policies.
3. Limited interoperability among frameworks that use different languages and policies.
4. Limited interoperability among frameworks that use different languages and policies.
5. Limited interoperability among frameworks that use different languages and policies.
6. Limited interoperability among frameworks that use different languages and policies.
7. Limited interoperability among frameworks that use different languages and policies.
8. Limited interoperability among frameworks that use different languages and policies.
9. Limited interoperability among frameworks that use different languages and policies.
10. Limited interoperability among frameworks that use different languages and policies.

Levying the power of the Semantic Web to reason across the various languages leads to WS-Policy (2005), TrustWeb (Gleave, et al. 2000), SPKI/SPKI (Goldberg, 2002), and role languages that are used in different frameworks. To this is another framework that will help the Web preserve maximum expressiveness for policy languages by allowing users to define policies in their own languages but still use the same mechanisms for deploying policy domains.

Best are Web-based policy management frameworks, which exploit the inherently distributed and open nature of the Web by allowing policies, meta-policies, and policy languages to be combined, annotated, and otherwise handled in a scalable, modular manner as are any Web resources. However, their policies and meta-policies, the policy languages used, and their relationships together form the policy networks. Role allows entities to define policy networks to be located on local or remote servers and provides a mechanism to reason over them. The ontologies and reasoning engines work with any policy language and domain knowledge defined in RDF/S, OWL, or supported role languages.

Though authorization is an important part of secure policy management, also does not enforce a unique policy. Rather the classes are planted into a role and the policies vary with the person. The ontologies and reasoning engines then work with any policy language and domain knowledge defined in RDF/S, OWL, or supported role languages.

We have developed examples that combine authentication and authorization and rely on simple cryptography techniques and other examples that use...
Purpose

- Develop a policy framework (Rein) that leverages the semantic web
- Allow users to define policies in their own language
- Provide mechanisms for reasoning over any supported rule languages
Contributions

- Web-based approach for representing and reasoning over policies for web resources
- Flexible sophistication or expressiveness of policies
- Provides unified mechanism for reasoning
- Supports compartmentalized policy development
- Self-describing framework
Access control model
Rein Ontology
Example Rein Policy Network
Rein Ontology

Rein Policy Network Ontology

Rein Request Ontology

Usable Security – CS 6204 – Fall, 2009 – Dennis Kafura – Virginia Tech
Sample Ontology
Reasoning Engine

- Accepts requests for resources
- Collects relevant information
- Answers questions about access rights
Using Rein

- Rein used by guard
- Rein used by client
- Hybrid approach
Implementation Requirements

- Reasoners for RDF-S
- Engine for supported rule language(s)
- Programming language capable of:
  - Accessing web
  - Working with chosen reasoners and engines
Sample Request
Rein implemented in Policy Aware Web
Discussion

- There is no discussion of time to process a request. The overhead for processing a complex request could be non-negligible.
- There are no security mechanisms in place. It seems pretty easy to perform a DOS attack, yet difficult to prevent it.
- Rein does not allow querying of what resources a requester has access to, only whether a requester can access a resource.