Semantic Web Policy Systems

Presented By: John Paul Dunning
A meta-control architecture for orchestrating policy enforcement across heterogeneous information sources

Jinghai Rao, Alberto Sardinha, Norman Sadeh
Carnegie Mellon University

Abstract

This paper introduces a meta-control framework for orchestrating policy enforcement across multiple information sources. The framework allows for the integration of different policies and their enforcement across heterogeneous environments. It provides a mechanism for managing policy conflicts and ensuring consistency in policy enforcement. The proposed framework is evaluated through a case study of policies enforced by different service providers.
Overview

- Context-sensitive security and privacy policies
- Decentralized trust management
- Challenges include:
  - sources of information vary from one principal to another
  - sources of information may vary over time
  - sources of information may not be known ahead of time
Contributions of Paper

- “Development of a semantic web framework and a meta-control model for opportunistically interleaving policy reasoning and web service discovery to enforce context-sensitive policies”
- Extension of XACML ontology
- Language independent system
XACML

“XACML is an initiative to develop a standard for access control and authorization systems...

XACML aims to achieve the following:

- Create a portable and standard way of describing access control entities and their attributes.
- Provide a mechanism that offers much finer granular access control than simply denying or granting access -- that is, a mechanism that can enforce some before and after actions along with "permit" or "deny" permission.”

Information Disclosure Agent (IDA)

- Policy Enforcement Agent (PEA)
- Controls access to information and service access through policies
- Uses policy enforcement
  - Control policies
  - Obfuscation policies
Information Disclosure Agent (IDA)

- Interact across various networks
- Encrypted traffic
- Language Independent (with interpreter)
Information Disclosure Agent (IDA)
Meta-Controller

- Monitors progress and determines the next step
- Cycle

Meta-Control

Housekeeping Module

Modules complete tasks
Meta-Controller

Query status information includes:

- A query status ID
- Status predicates
- A query ID and query element ID
- A parent query status ID
- A time stamp
## Meta-Controller

<table>
<thead>
<tr>
<th>Sample Status Predicates</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(1) communication status predicates</strong></td>
<td></td>
</tr>
<tr>
<td>Query-Received</td>
<td>A particular query has been received.</td>
</tr>
<tr>
<td>Sending-Response</td>
<td>Response to a query is being sent</td>
</tr>
<tr>
<td>Response-Failed</td>
<td>Response failed (e.g. message bounced back)</td>
</tr>
<tr>
<td><strong>(2) query status predicates</strong></td>
<td></td>
</tr>
<tr>
<td>Processing-Query</td>
<td>Query is being processed</td>
</tr>
<tr>
<td>Potentially-deadlocked-query</td>
<td>Used to flag queries that may correspond to possible deadlocks</td>
</tr>
<tr>
<td>Query-Decomposed</td>
<td>Query has been decomposed (into primitive query elements)</td>
</tr>
<tr>
<td>Query-Succeeded</td>
<td>All query elements are available and cleared. Ready to send response.</td>
</tr>
<tr>
<td>Query-Failed</td>
<td>Some query elements are not available or cleared.</td>
</tr>
<tr>
<td>All-Elements-Available</td>
<td>All query elements associated with a given query are available (i.e. all the required information is available)</td>
</tr>
<tr>
<td>All-Elements-Cleared</td>
<td>All query elements have been cleared by relevant access control policies</td>
</tr>
<tr>
<td>All-Elements-Sanitized</td>
<td>All query elements have been sanitized according to relevant obfuscation policies</td>
</tr>
<tr>
<td>Query-make-deadlock</td>
<td>The incoming query may result in an endless loop. According to different meta control rules, the IDA may respond a failure to query sender, or consult the user to handle the problem.</td>
</tr>
</tbody>
</table>
Policy Reasoner

- Evaluating relevant policies
- Return policy decisions
- Modules:
  - Query Decomposition Module
  - Access Control Module
  - Obfuscation Module
Information Collector

- Gathering facts

- Modules:
  - Local Information Reasoner
  - Service Discovery Module
  - Service Invocation Module
  - User Interface
Service Discovery and Invocation

- IDAs are constantly sending queries and results back and forth
- Multiple queries between IDAs
- Node deadlock is possible and avoidable
  - Time outs
  - Query dependency graphs
Example Scenario

- Bob is an employee of SATElectronics Corporation
- Bob contracts to United GenSat
- Bob wants the schedule for deployment of SAT 777 from United GenSat, which is a product he has been working on.
Example Scenario

A query received. Bob asks the production schedule of "SAT 777".

Query Element Status
(query-received (sender bob) (ask (schedule "SAT 777" ?X)))

The next step is to evaluate whether Bob can access the required information.

Query Element Status
(clearance-needed (User bob) (element (schedule "SAT 777" ?X)))

This invokes the Policy Reasoner, which is the Access Control Module in our example. After applying the access control policies, two elements are required:

Access Control Status
(query-element (hasPreferredSupplier UnitedGenSat SATElectronics))
(query-element (hasAuthorizedEmp SATElectronics Bob))

The first query element can be known from local KEB, while the second can not.

Information Collection Status
(element-not-known-locally (hasAuthorizedEmp SATElectronics Bob))

The meta controller starts a service discovery task.

Information Collection Status
(element-needed-service (hasAuthorizedEmp SATElectronics Bob))

A service is found.

Information Collection Status
(service-identified (name Auth service) (endpoint serAg) (query (sender UnitedGenSat hasAuthorizedEmp SATElectronics ?F)))

Bob is an authorized employee according to the result of the service.

Information Collection Status
(service-response-available (service-name Auth service) (hasAuthorizedEmp SATElectronics Bob))

After knowing all the required information, the access is granted.

Access Control Status
(element-cleared (user Bob) (element (schedule "SAT 777" ?X)))

After all query elements are finished, the query is finished, and the answer is sent back to the requester.

Query Element Status
(response-sent (receiver Bob) (schedule "SAT 777" Schedule1))
Beyond Access Control Policies
Q&A

- How easy are the policies to create/update/delete?
- What is the overhead of this system VS a standard form of authentication?