perMIT from MIT

Respondent

Paul Hill, MIT Consulting Technical Architect

Architect and Project Manager for the perMIT project

Goal/Problem Space

perMIT provides a consistent way to store and maintain access rules for other applications, such as SAP. Web applications serve as a common tool for users in offices and labs to maintain access rules (privieges) for their systems. Privileges are stored in perMIT's central database. Web applications or a web service can be used to display, create, or modify them. PerMIT does not enforce the access rules that it maintains. In XACML terminiology perMIT is primarily a PAP (Policy Administration Point). However, there are web service interfaces that also enable its use as a PDP (Policy Decision Point). It never functions as a PEP (Policy Enforcement Point). perMIT is an open source evolution of the MIT Roles system that has been in production at MIT for well over a decade.

Features

- perMIT provide a rich, RDF-like, model for managing privileges and meta-privileges.
- A categorization model allows developers to decide if privileges should be common across a set of related applications, or specific to an individual application.
- Starting and ending time bounds can be applied to any defined privilege.
- Privileges can be explicitly granted to an individual.
- Privileges can be populated via a data driven rules evaluation engine. (implied authorizations)
- An inheritance model is provided via the use of a hierachical scoping model (aka Qualifiers).
- Audit trails are provided and a rich set of audit reports can be generated from the system
- The maintenance of privilege definitions may be distributed and delegated throughout an organization
- Supports multiple master organizational hierarchies (e.g. data can be viewed and transformed between HR's view of the organization, Finance's view of the organization, or any other defined hierarhcy that provides a view into the organization.)

Technology Stack

- MySQL database
- sample data feed scripts written in perl
- Perl CGI web applications
- SOAP based web service implemented in Java
- WebUI application implemented in Java which uses the SOAP based web service

Identity Services

Please indicate which of the following identity services you consume, produce, or broker/convey.

- **Consume**: Your project uses the services described. For example, you use identification information to determine which person you are dealing with, and you are a client to an authentication interface to confirm the person's identity.
- **Produce**: Your project provides the services described. For example, you provide facilities to manage groups and can write them out to LDAP.
- **Broker/Convey**: Your project serves as a middleman, taking data from a producer and providing it to a consumer. For example, you verify authentication information and then generate a SAML assertion.

Managed Information	**Consume?**	**Produce?**	**Broker /Convey?**
Privileges	Y(1)	Y	Y
Roles	Y(2)	Y(2)	N
Groups	Y(3)	N(4)	N(4)
Attributes	Y(5)	N	N
Identification	Y(6)	N	N
Defined Interfaces	**Consume?**	**Produce?**	**Broker /Convey?**
Authentication	Y(7)	N	N
Attributes	N	N	N
Permissions	Y	Y	Y
Provisioning	Y(8)	N	Y(9)
Authorization	Y(10)	Y	Y
Subjects	Y	N	N

Other	Consume?	Produce?	Broker /Convey?

1 - perMIT uses perMIT to manage its own privileges. perMIT has a category for meta-ASPEC definitions.

2 - Although perMIT does typically define and manipulate "roles" in the RBAC sense it does deal with "roles" in two different ways in special cases. It does have a concept of "primary authorizer"; the primary authorizer role may be assigned to people with a departmental or school scope. People that have this role assigned to them may grant authorizations that have the same, or subsideary scopes, to anyone within the perMIT namespace. This role crosses some category boundaries.

The other case where perMIT blurs the line regarding Roles is in the area of implied authorizations. The implied authorization subsytem lets one create a set of business rules and define data sources, that when evaluated, will cause privileges, in the form of ASPECs to be defined.

3 - It is possible to use group memberships (from other systems) as input into a rule evaluation which creates implied authorizations.

4 - Some initial work has been done to show how perMIT ASPECs can be exposed as traditional groups in other systems. We have one whitepaper (currently a draft) that discusses this topic. However, this technique is not yet being used in production on any systems.

5 - Attributes from other system may be used as input to a rule evaluation for implied authorizations. perMIT does not consume attributes directly, nor does it issue them.

6 - perMIT constrains what identifiers may be entered. It does not allow a user to enter an unknown identifier. This helps to reduce bad or unusable authorizations from being defined. Since a lookup is also done in the normal GUI, the person doing data entry sees more information than just a username, this also helps to reduce bad data inputs.

7 - Strong authentication is used to access the perMIT system and its services.

8 - Some of our account provisioning systems do automatically feed certain tables with perMIT. However, the provisioning process does not cause authorizations to be defined automatically. We consider that to be outside the scope of provisioning, instead that is part of implied authorizations.

9 - Since provisioning within the identity management system may also cause some metadata to be created, it is possible that a normal account provisioning process will result in an implied authorization being granted, you could say that we do some brokering of provisioning. (e.g. a faculty member being hired in Biology will automatically result in the person being hired having the privilege necessary to access some third party database content.)

10 - perMIT does not consume external authorization data. However, perMIT does consume its own authorization data for access to perMIT features and data.

Standards and Interfaces

Issues and Challenges

perMIT is an ongoing project at MIT. The first phase of the project is nearing completion. We have begun planning the next phase which is about deploying perMIT at MIT and starting to replace the existing Roles system.

One area that is subject to change is the web service interface. The current interfaces were designed by MIT IS&T for use at MIT. We are interested in seeing if well designed service interfaces for authorization managemenet emerge, either as a standard, or a de facto standard. We are keeping an eye on the work coming out of Kuali Student.

More Information

https://wikis.mit.edu/confluence/display/PERMIT/perMIT+project