

Semantica Enterprise Security Model

Tim LaMarca

November 17, 2003

Abstract

Semantica Enterprise is Semantic Research Incorporated's flagship product for knowledge capture, management and transfer based on Semantic Networking Theory.

The Enterprise Server is collaborative and expected to handle both sensitive and confidential data. As such, it must implement a security model to control access to and use of the content it contains.

In this paper, we discuss such a model.

Contents

1 Overview

Security is a chief concern to those handling content. Sensitive material is not meant for prying eyes.

Within an enterprise, security is typically handled at many different levels. Among those may be physical security, network security, datastore (database) security and others. Essentially, security is designed to allow some people in and keep others out.

The Semantica security model is based on three referential sources. Namely: databases, operating systems (UNIX, Windows), and existing collaborative tools. The idea behind the system is to control who is allowed to access what content.

As a matter of practicality, the system can be implemented in sections and need not exist in its entirety in order to be functional. This ensures that some minimally functional version can be created and extended as more granularity is desired.

The basis of the system used by Semantica Enterprise is the Java (JAAS / PAM based) authentication and authorization model. In this model, there are two distinct security functions. The first (authentication) is the identification of the agent that wishes to access the system and the successful presentation of their credential(s) for the purpose of verifying their privilege to access the system. The second (authorization) is the process of determining which access rights the agent has that enables them the ability to manipulate, view or create content within the system.

Authorization is handled by the JAAS and J2EE framework that exists for this processes. In essence, this is a plug-able model that authorizes agents based on their credentials against a database, ldap directory, PKI infrastructure, active directory, or any other existing or developable security mechanism and store. This process serves to identify both the agent and any security roles that the agent possesses. A default Semantica Enterprise installation requires user name and password based authentication verified against a RDBMS data store.

Once authorized, Semantica Enterprise exposes an abstract semantic based security model for protecting content within the system. This model allows for the protection of content using a system that grants or denies access to content. Access can be as broad as the network level and as granular as the individual element level within the network. A default Semantica Enterprise installation supplies an implementation of the authorization interface that uses a semantic network to persist security information.

An example of system use follows:

- 1) A user would like to open a semantic network to review add, edit or perhaps delete certain content elements.
- 2) An object is created to represent the person wishing to access the system and their security credentials are associated with the object.
- 3) The security object is presented to the authorization subsystem for authentication. If the subsystem denies the person, no further action is or can be taken. If the subsystem allows the person, the subsystem associates roles with the individual and establishes a unique identity for

them as well as provides this identity and role information to the enterprise application server.

4) The person then attempts to manipulate content within the system.

5) Semantica enterprise requests the identity of the calling agent (person) from the application server and passes the identity of the user along with a semantic description of their attempted action to the abstract authorization interface.

6) The abstract interface then asks a concrete implementation (in this case a Semantic Network security Policy) to grant or deny the users privilege to follow through on such an action.

7) Based on the response of the security sub-system, the transaction is allowed to continue, or aborted.

2 Design

There are three core components to the design of the security model. These are: 1) Elements, 2) Permissions and 3) Principals (users/agents). Groups and Roles as are also useable as a convenience but not required.

2.1 Elements

Elements are components of a semantic network and therefor entities in the Semantica system. Both elements and their functionality are protected.

The current list of elements is :

- Concepts
- Knowledge Objects
- Instances
- Networks
- Relations

2.2 Permissions

For each element there is an item of functionality with a corresponding permission.

For example, if a concept can be edited, then there is an associated permission.

Permissions can be set on individual elements or collections of elements.

Permissionable actions are:

- Add
- Edit
- Get
- Remove
- Size

Permissions can be inherited. That is, if a user has the right to edit concepts in a network, the user does not need to have an individual edit permission set for each concept. However, the permission could be denied for specific elements.

Permissions can be held by any Principal. Principals may represent:

- Users
- Groups
- Roles

3 Enterprise Integration

Abstract classes providing security functionality belong to the *semantica.security* package. The semantic network implementation is contained in the *semantica.security.sn* package.

There are three key interfaces:

- *SPrincipal* implements `java.security.Principal` and represents a single agent or a key part of the agent (user) that wishes to access the system.
- *SPermission* extends `java.security.Permission` in order to implement a permission scheme consistent with the Semantica permission operations supported for *SElement* classes.
- *PrincipaledSecurityManager* partially models `java.secutiy.SecurityManager` and enables the lookup of a permission for a set of related principals.

The semantic network implementation of these interfaces relates principals to Permissions as instances within the security network.

For example:

```
{user001, add, semantica.network.*}
{user002, edit, semantica.network.concept*}
{user003, remove, semantica.network.*}
{user004, get, semantica.network.jklg-jklp-jkl-asdf.*}
{user004, not get, semantica.network.jklg-jklp-jkl-asdf.wert-adff-hjki-pycb}
```

[where xxxx-xxxx-xxxx-xxxx are unique IDs that link to actual elements in the referenced network or identify the referenced network itself]

Semantic Research is revolutionizing the way we visualize, store and communicate knowledge through the practical application of semantic network theory. Semantic networking is based on over thirty years of research in artificial intelligence, cognitive psychology, memetics and learning theory, and has been independently proven to be significantly more effective in the transfer of knowledge from expert to learner.

Semantic Research Inc. reserves the right to change specifications and other product information without prior notice. This publication could include technical inaccuracies or typographical errors. Semantic Research Inc. provides this publication "AS IS" without warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Semantic Research, Inc.
1055 Shafter Street
San Diego, California 92106 U.S.A.

Copyright 2003 Semantic Research Inc. All rights reserved.