A Flexible Security System for Enterprise and e-Government Portals

Torsten Priebe, Björn Muschall, Wolfgang Dobmeier, and Günther Pernul

Department of Information Systems
University of Regensburg, D-93040 Regensburg, Germany
{torsten.priebe,bjoern.muschall,guenther.pernul}@wiwi.uni-regensburg.de
wolfgang.dobmeier@gmx.de

Abstract. Web-based systems like enterprise and e-government portals pose special requirements to information security. Today's portal platforms provide some security functionality, mainly targeting at supporting a single-sign-on for the underlying applications. We argue that single-sign-on is not sufficient, but rather a mature security service is needed as a central authorization instance. As access control is needed on different levels of a portal architecture, only this allows an integrated approach to security management. We present CSAP (Communication Security, Authentication, and Privacy), a flexible security system for enterprise and e-government portals. CSAP was originally developed within the EU-funded research project "Webocracy". Meanwhile, various enhancements to CSAP have been made, which are being discussed in this paper. The major enhancement is a Metadata-based Access Control facility (MBAC) which allows more flexibility in highly open and heterogeneous systems. We use CSAP within two portal prototypes, one in an enterprise one in an e-government context, which are being presented as case studies.

1 Introduction

The use of Web-based portals has become a popular approach for providing an integrated user interface for different information sources and application systems in an intranet as well as Internet context. This is particularly true for an enterprise environment where employees, customers or suppliers are accessing such portals. However, also the public sector is employing similar technologies as part of its e-government initiatives. While portals integrate the applications by combining several user interface components (so called portlets) on a single portal webpage the backend integration has to be performed by other means. Technologies like Enterprise Application Integration (EAI) are evolving here. At the same time it is being recognized that in such global and integrated environments also (and in particular) security aspects cannot be properly addressed in an isolated fashion, either. Portals are required to provide a single-sign-on facility to avoid the necessity for users to authenticate themselves multiple times. However, single-sign-on is not enough. The administration of users, roles, etc. and their permissions on security objects also needs to be supported in an integrated way.

F. Galindo et al. (Eds.): DEXA 2004, LNCS 3180, pp. 884–893, 2004.

As an answer to these issues, we present CSAP, a flexible security system for enterprise and e-government portals. The CSAP security module was originally developed within the EU-funded research project "Webocracy" (IST-1999-20364). Meanwile, the Metadata-based Access Control (MBAC) model presented in section 3.2, as well as its implementation and further enhancements to the CSAP security module have been developed within the project "SeWiss", funded by the Bavarian Research Cooperation for Information Systems (FORWIN)[1].

The remainder of this paper is organized as follows: Section 2 introduces portal systems and gives an overview on security aspects, identifying authorization and access control as major issues. Section 3 discusses two access control models, the well established Role-based Access Control model (RBAC) as well as the novel Metadata-based Access Control model (MBAC). Both have been implemented within our CSAP security module which is being used in two portal prototypes. These are covered by section 4. Finally, section 5 concludes the paper by discussing possible future work.

2 Security in Portal Systems

Enterprise portals focus on corporate information and services being provided to employees (B2E, business-to-employee portals) or customers and suppliers (B2C/B2B portals). The goal is to provide the user with a consolidated, personalized user interface to all information he needs. Recently the term enterprise knowledge portal is more and more used instead of enterprise information portal. Advanced techniques try to help the user with accessing the right information at the right time. This implies the support of organizational learning and corporate knowledge processes.

In an e-government context similar techniques are being utilized to provide citizens with administrative information and to allow them to participate in democratic processes (G2C, government-to-citizen portals). In addition, G2B (government-to-business) portals support processes like public tendering.

One important property of such portals (in both, an enterprise and public context) is their openness. The number of users accessing the portal – consuming or publishing content – is usually very high. The same is true for the number of information objects (e.g. documents), possibly containing sensitive information. Obviously, these conditions lead to special security requirements. Like for most information systems the major security services needed are user identification and authentication and authorization and access control. When a user has been identified (e.g. by a username) and authenticated (e.g. by a password) authorization deals with managing the permissions which define which users can access certain information or system functions. Access control checks these permissions at run time avoiding unauthorized access.

Access control in portals has to be applied on different levels. Firstly, portal platforms provide access control on a structural level, defining which users can

[1] `http://www.forwin.de`.

access which parts of the portal (i.e. which portlets can be viewed, configured, etc.). On the other hand, access control for the actual content, e.g. individual documents, is performed within the portlets. Often this access control is even enforced by the underlying applications rather than the portlets themselves which represent only user interface components. The portlets usually use only the authentication (login) feature of the portal to provide a single-sign-on capability. However, a truly integrated approach to authorization and access control is highly desirable in order to be able to provide centralized and integrated security management.

3 Access Control Models

A security system that can be used portal-wide (and possibly even by the underlying applications) needs to be flexible enough to support different environments. For many applications with structured security requirements (especially in an organizational context) the well-established Role-based Access Control model (RBAC) is very suitable and can be seen as a de-facto standard. Section 3.1 gives a short overview of this model.

However, in rather open and heterogeneous applications the security requirements are hard to model in form of structured role hierarchies. To address such environments we have developed a more flexible Metadata-based Access Control model (MBAC) which will be described in section 3.2. We argue that a flexible security system for enterprise and e-government portals should support both role- and metadata-based access control.

3.1 Role-Based Access Control

The concept of Role-based Access Control (RBAC) [7] has evolved to the defacto standard in enterprise multi-user systems, involving a large (but structured) number of users with different rights and obligations as well as a large amount of sensitive data. In contrast to simpler earlier access control approaches RBAC simplifies the administration of access permisions by means of roles which can be derived from the organizational structure.

Experiences in organizational practice show that the definition of roles as part of the organizational structure can be seen as relatively stable while the assignment of users to roles in comparison changes more frequently. This assignment can be abolished without affecting the definition of a role itself while on the other hand the role's definition can be modified without interfering with the assignment. Furthermore, by assigning different users to the same role and different roles to the same user, redundancy can be avoided.

In [7] a standard for Role-based Access Control has been proposed. The RBAC reference model is divided into submodels which embody different subsets of the functionality.

Core RBAC covers the essential aspects such that permissions are assigned to roles and roles are assigned to users. It comprises five basic element sets. Users

are active elements that can be human beings as well as software artefacts. Roles correspond to fields of activities of users. These activities are bound to permissions needed to carry them out. Only such permissions should be assigned to a role that are absolutely necessary to fulfill the corresponding activities. A permission is a certain operation (e.g. read) that can be executed on a certain object. Objects and operations reflect arbitrary system objects and methods at different levels of granularity. A session represents the context in which a sequence of activities is executed.

In addition to Core RBAC, Hierarchical RBAC introduces a partial order between the roles, an inheritance relation where senior roles acquire the permissions of their juniors and junior roles acquire the user membership of their seniors. There may be general permissions that are needed by a large number of users and consequently may be assigned to a more general role. Finally, Constraint RBAC assumes that there are relations or exclusions between some fields of activity and allows to define separation of duty constraints to enforce conflict of interest policies.

3.2 Metadata-Based Access Control

The Role-based Access Control model presented in the previous subsection simplifies the administration of authorizations. However, for very large open systems such as digital libraries, enterprise or e-government portals, or hospital systems, the role hierarchies can become very complex. When the number of protected objects also increases, a manual assignment of authorizations becomes very expensive and error-prone. Furthermore, in many situations access depends on contents of an object and the environment the subject is acting in. In these applications we need to deal with users not previously registered. The Metadata-based Access Control model provides a more convenient and efficient way to manage access rights for these situations.

The basic idea is to utilize (possibly dynamic) properties of subjects and objects as the basis for authorization, rather than directly (and statically) defining access rights between users, roles, and objects. On the user side, an attribute could be his position within an organization, quite similar to a role. Especially for external users however, acquired credentials (e.g. subscriptions, customer status) or attributes such as age or shipping address may need to be used instead. For the security objects, the content, e.g. of documents, can be described by means of metadata. Such metadata elements should be used for authorization purposes.

Two primary directions of research have evolved. The first derives from research on security for digital libraries. [1] propose a Digital Library Access Control Model (DLAM), which defines access rights according to properties associated with subjects and objects. The second important direction of research has its origin in the area of public key infrastructures (PKI) and is based on the use of certificates for authentication. A widespread standard for certificates is X.509 [8], which enables the user to employ his private key for authentication, while the respective addressee is using the certified corresponding public key for

checking the claimed identity. In addition to the public key, also other attributes can be assigned to the owner of a certificate. [2] proposes to use these attributes for authorization and access control purposes.

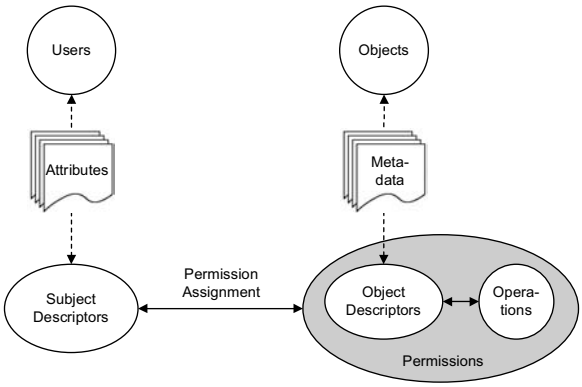

Fig. 1. MBAC model

Figure 1 shows the elements of the Metadata-based Access Control model. A subject is represented by a number of attributes (also called credentials). Similarily, an object (i.e. a resource to be protected) is described by a number of properties. The permissions are no longer assigned between subjects and objects but between subject and object descriptors. A subject descriptor consists of a number of attribute qualifiers (e.g. age > 18, ZIP code begins with "93") and can be seen as a "virtual subject" that can possibly relate to multiple real subjects. The same is true for the object descriptors. Here qualifiers are defined on the basis of object properties, e.g. concerns a certain project, published by a certain publisher.

The MBAC model has been described in detail as a security pattern in [9]. In order to reach more flexibility, enhanced (more complex) versions of the MBAC model have been developed. These enhancements have also been described as security patterns reusing the basic MBAC pattern.

The first enhancement concerns the principle of "least privileges" and introduces a session concept as known from RBAC. The user has the option to activate only a subset of the attributes assigned to him within a session. This might also help to enhance user privacy. A user will potentially only want to disclose personal data (e.g. age or sex) if it is indispensable for the task he intends to perform[2]. For example, an e-commerce system selling adult entertainment needs to know the age of a customer before granting him access to its products. If attributes are assumed to be centrally stored and managed, users have only

[2] See also the controlled disclosure of personal properties on the Internet in P3P (Platform for Privacy Preferences) [12].

limited control over the use of their personal data. The idea of storing user attributes in attribute certificates rather than in a central database may help to overcome this issue.

A second enhancement to the MBAC model introduces predicates. While the basic MBAC model only allows the use of constant values to define subject and object descriptors, predicates allow comparisons between attribute and property values. For example an object within a digital library might have the property "publisher". A user might pay a subscription fee for accessing content by a certain publisher, represented by an attribute "subscribed for". A predicate could define that a certain permission can only be utilized if the user's "subscribed for" attribute matches the "publisher" property of the object to be accessed.

4 Implementation and Prototypes

4.1 Enhanced CSAP Security Module

As mentioned before the authors have been involved in the EU-funded project "Webocracy" in which the "Webocrat" e-government portal has been developed (presented in section 4.2). Within this project our primary task was to implement a security management module called CSAP (Communication Security, Authentication and Privacy) whose purpose is to form the basis of trust for the whole system by offering "practical and consistent" security. This has been achieved by providing services for user identification and authentication, access control and authorization, auditing, and session management.

CSAP was designed to satisfy differing requirements by providing alternative implementations of security services via a plug-in concept. The original version of CSAP developed within "Webocracy" [4] implements a password-based authentication scheme and an RBAC access control capability. In order to be more flexible and applicable for a wider range of applications, we have enhanced CSAP to provide also an MBAC access control service. An authentication mechanism based on X.509 certificates [8] is subject to current work (see section 5). The goal is to provide a security system that is suitable as a central security component for enterprise and e-government portals addressing the demands stated in section 2.

The overall architecture of CSAP is shown in figure 2. It consists of two layers: the service layer where the actual security services are implemented, and a data layer that defines a generic interface in order to access the security metadata independently of the storage technology of choice. Note that the dashed lines in the figure symbolize components that have not yet been fully implemented.

CSAP is implemented as a Java class library but has also been designed for being used in a distributed environment. That is, it offers RMI access to its services via the API. All data objects like a user or a session object can be serialized and sent over a network as parameters to the remote services. This allows the distribution of services to different machines, e.g. for balancing system load. Moreover, it allows CSAP to be accessed remotely by different applications at the same time as a single central security instance.

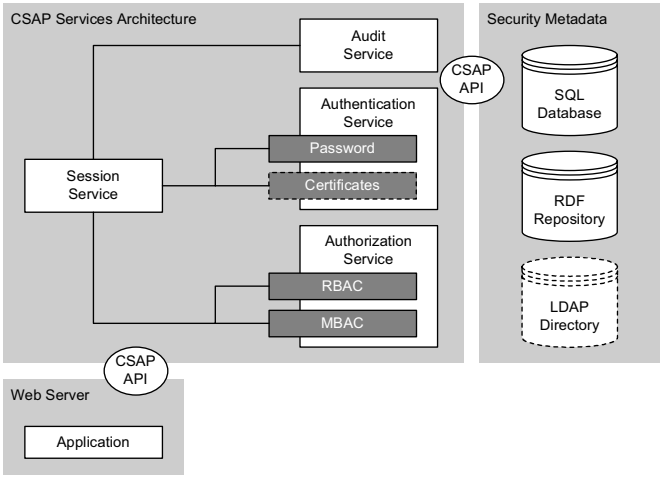

Fig. 2. CSAP architecture

Another aspect in this context is the multiple use of sessions. After a user has been authenticated, a session object is created. This session can be used by different applications independently of each other. This way, a session that is globally established by the authentication dialog of a portal can be reused on various levels by other components, i.e. portlets or underlying application systems, providing a single-sign-on facility.

As mentioned earlier, we extended CSAP by an MBAC access control service. In the current version the implementation covers the MBAC model in its basic form, i.e. it does not support predicates and sessions. The implementation is done as an additional authorization service which can be accessed via corresponding API methods. It can be used together with the RBAC service, thus multiple access control models are available in a single instance of CSAP. This facilitates a deployment in environments with the need for different access control methods.

The implementation is based on the MBAC pattern presented in [9]. Its structure was converted into an ER model and integrated into the CSAP database design for the security metadata. To improve the performance when a large number of access control requests needs to be executed in a short time (e.g. by a search engine, where the search results are to be checked before being presented to the user), we implemented the option to precalculate and store all subject descriptors that match a certain subject. We plan to expand this caching approach to the object descriptors and objects, respectively.

4.2 e-Government Portal "Webocrat"

As mentioned before the authors have been involved in the research project "Webocracy". The "Webocrat" portal developed within the project is an example of

innovative use of state-of-the-art technologies to provide citizens, businesses, and government agencies with more convenient access to governmental information and services. The project's main goal is to develop and investigate Internet and Web technologies which increase the quality of the offered services, improve efficient and transparent access and consequently provide greater opportunities to participate directly in democratic processes.

The "Webocrat" e-government portal provides a central user interface, which integrates several application systems like discussion and opinion polling rooms, components for knowledge management, tools for content management and publishing, and finally systems for intelligent retrieval of information and explorative analysis of data. All modules share common access to the central CSAP security module. For details see [6].

Within "Webocrat" we are now evaluating the enhancements (e.g. the MBAC access control service) that have been made to the original CSAP module.

4.3 Enterprise Knowledge Portal "INWISS"

A prototype of an integrative enterprise knowledge portal called "InWiss" [10] is in development within another project of the authors' group. In this project we integrate various knowledge management components through a portal in such a way that user actions within one portlet can be propagated to others. For example, if a user navigates in an OLAP report through slicing/dicing or drilling, a search engine can evaluate the user's current query context to offer relevant documents from the Internet or a document management system. We use context-related metadata based on RDF [11] for that purpose. An overview of the architecture is provided in figure 3.

The Apache Jetspeed framework[3] provides the basis for the portal user interface for which we developed an OLAP portlet to display OLAP reports accessing a data warehouse, a search portlet to perform semantic (metadata-based) searches, and a news portlet as an example for native portal content.

The CSAP Security Module is the central instance where users are authenticated and access control is performed, triggered from various layers of the architecture. We modified the login service in Jetspeed to use CSAP to perform the desired authentication and create a user session. The session data is put into the Jetspeed environment, ready to be used by other components of the portal. We also connected the user administration of Jetspeed to CSAP in a way that the management of a user's properties (e.g. password) or adding new users can be done via the Jetspeed security administration portlet. This way, one security module can be used for almost all components of the portal, avoiding heterogeneous and decentralized security administration.

The semantic search engine provides a possibility to (explicitly and implicitly) search for documents on the basis of metadata and an enterprise ontology. For access control, we use our metadata-based approach as described in section 3.2. While the metadata for the subjects (i.e. the user attributes) is kept in the CSAP

[3] `http://jakarta.apache.org/jetspeed/`.

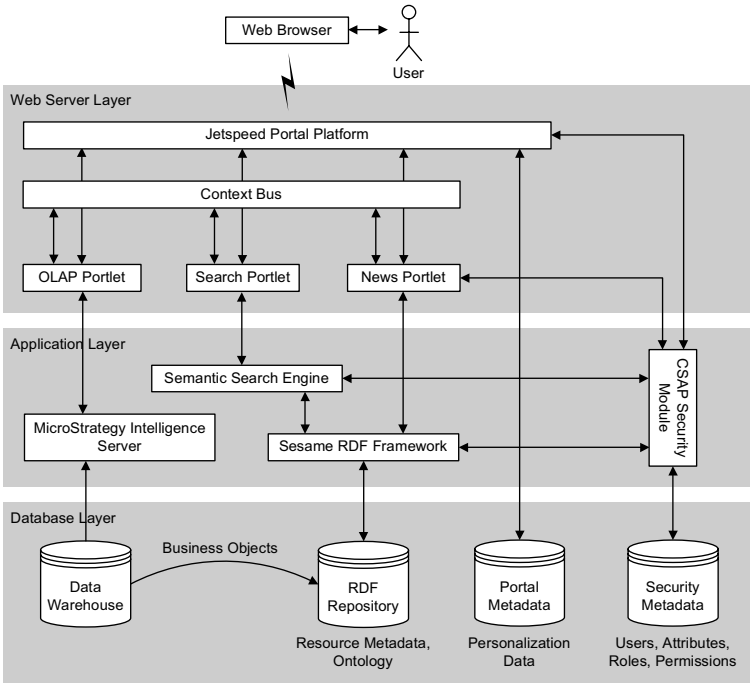

Fig. 3. "InWiss" enterprise knowledge portal prototype architecture

database, the object metadata is stored in a repository that is maintained by the Sesame RDF framework[4] [3].

5 Conclusions and Future Work

We have presented CSAP, a flexible security system for enterprise and e-government portals. We discussed two sample prototype portals that utilize the system as well as several enhancements to the system that have been made since the original version that has been implemented within the "Webocracy" project. The main enhancement was the Metadata-based Access Control (MBAC) facility. Some further technical improvements we are currently working on are an XML-based import/export facility for the security metadata and advanced analysis mechnisms for the audit trail.

As mentioned before an important motivation for an integrated security system like CSAP is to have a central point for security management. Users and security objects, as well as authorization entities like roles and subject or object descriptors, should be managed centrally. A first approach for such a security

[4] `http://www.openrdf.org`.

administration console has been presented in [5]. We are currently porting this console to a portlet implementation which allows integrating it into the portal user interface. The support for RDF-based object metadata has been another step towards application interoperability. On the user side we are working on supporting an LDAP directory for storing the users and user attributes respectively.

A further enhancement concerns the support for X.509 certificates for authentication as well as for attribute storage. PKI support, standardized interfaces (like LDAP) and additional security services (like timestamping) might turn CSAP into a service component suitable also for distributed authentication and authorization infrastructures (AAI).

References

1. Adam, N.R., Atluri, V., Bertino, E., Ferrari, E.: A Content-based Authorization Model for Digital Libraries. IEEE Transactions on Knowledge and Data Engineering, Volume 14, Number 2, March/April 2002.
2. Biskup, J.: Credential-basierte Zugriffskontrolle: Wurzeln und ein Ausblick. 32. Jahrestagung der Gesellschaft für Informatik e.v. (GI), Dortmund, Germany, September/October 2002, pp. 423-428.
3. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Proc. of the First International Semantic Web Conference (ISWC 2002), Sardinia, Italy, June 2002.
4. Dridi, F., Fischer, M., Pernul, G.: CSAP an adaptable security module for the e-government system Webocrat. Proc. of the 18th IFIP International Information Security Conference (SEC 2003), Athens, Greece, May 2003.
5. Dridi, F., Muschall, B., Pernul, G.: Administration of an RBAC System. Proc. of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Big Island, Hawaii, USA, January 2004.
6. Dridi, F., Pernul, G., Sabol, T.: The Webocracy Project: Overview and Security Aspects. In: Schnurr, H.-P. et al. (Eds.): Professionelles Wissensmanagement: Erfahrungen und Visionen. Aachen, Germany, 2001.
7. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D., and Chandramouli, R.: Proposed NIST Standard for Role-based Access Control. ACM Transactions on Information and Systems Security, Volume 4, Number 3, August 2001.
8. ITU-T Recommendation X.509: The Directory – Public Key and Attribute Certificate Frameworks. 2000.
9. Priebe, T., Fernandez, E.B., Mehlau, J.I., Pernul, G.: A Pattern System for Access Control. To appear in: Proc. of the Proc. 18th Annual IFIP WG 11.3 Working Conference on Data and Application Security, Sitges, Spain, July 2004.
10. Priebe, T., Pernul, G.: Towards Integrative Enterprise Knowledge Portals. Proc. of the Twelfth International Conference on Information and Knowledge Management (CIKM 2003), New Orleans, LA, USA, November 2003.
11. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`
12. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, 2002. `http://www.w3.org/TR/2002/REC-P3P-20020416/`