Asian Journal of Information Technology 5(6): 603-608, 2006

Stimulating Knowledge Sharing Through Online Collaboration

Ismail Ahmad and Hong Tat Ewe
Faculty of Information Technology, Multimedia University,
63100 Cyberjaya, Selangor Darul Ehsan, Malaysia

Abstract: This study proposes a framework called C-SKAR which supports interaction and collaboration between two or more knowledge sharing systems in a secure fashion. The security architecture deployed within C-SKAR is capable of protecting collaborative knowledge acquisition and retrieval through the enforcement of site authenticity, access entitlements, dynamic constraints and access rules. Using this approach, existing online knowledge sharing systems can leverage on the knowledge accumulated by other knowledge sharing systems and therefore increase their local content efficiently and systematically. This would contribute to a healthy growth of collaborative knowledge sharing on the Internet that could in turn result in broad positive impacts on overall knowledge transfer between humans and the society.

Key words: Knowledge, interaction, collaborative knowledge

INTRODUCTION

The emergence and subsequent improvements in internet technologies have improved the way humans communicate. Knowledge typically owned by a particular organization can now be shared across many other organizations seamlessly. Disparate heterogeneous knowledge sharing systems can now intercommunicate with one another via automated channels and open standards interfaces. The objective of collaborative knowledge sharing is to allow all knowledge sites to interconnect with one another and share knowledge objects. The rapid growth in online knowledge sharing systems adopting open standards interfaces such as Extensible Markup Language (XML)[1] alongside protocols such as Simple Object Access Protocol (SOAP)[2,3] has facilitated this. Improvements made in digital data processing which resulted in the ability to store knowledge objects in the form of documents, images, audio and video streams have positive impacts on knowledge sharing. However, with the increasing reliance on online knowledge sharing technologies, there is also an increasing danger that the knowledge objects stored in digital form are subjected to concerted malicious and possible criminal attacks. Digitized knowledge objects can become easy prey to tampering and modifications during transmission and sharing thus defeating the original intent of knowledge sharing. In facilitating collaborative online knowledge sharing, systems design must be capable not only of protecting the integrity of the knowledge objects but must also be capable of offering a detection and recovery mechanism to make sure that knowledge processing resources continue to perform correct processing operations at all time. In this study, we present an approach to stimulate knowledge sharing through online collaboration.

RELATED WORK

Security in knowledge sharing systems aims at ensuring that knowledge objects are strongly protected and precise control to which users will be allowed to have access to each piece of the object within the systems[4]. Additionally, knowledge objects stored in systems' repositories must be guaranteed to be free of any form of unauthorized modifications[5]. This requirements can be simplified into four key aspects namely; usage auditing, user rights entitlements, encryption and repository access.

Usage auditing aims at tracking every single user interaction with the knowledge sharing system and once a user accesses a particular knowledge sharing system, their entire path of activity can be traced. User rights e ntitlements safeguards the system from unauthorized usage and one way to do this is through role assignment for each user. Encryption aims to ensure that all knowledge objects accessed during both storage and transmission are protected from being stolen. Repository access aims at ensuring that knowledge objects are only accessed based on accredited access rules and conditions. For collaborative knowledge sharing, additional security measure must be introduced when

Corresponding Author: Ismail Ahmad, Faculty of Information Technology, Multimedia University, 63100 Cyberjaya, Selangor Darul Ehsan, Malaysia

allowing external knowledge systems to access the local objects.

Many solutions exist to address security issues[7,8] in information systems but the mechanisms and frameworks are not specifically targeted to knowledge sharing systems. Nonetheless, the myriad of solutions offered can be synthesized and a dapted for knowledge sharing frameworks. As such, solutions offered by areas that have been rigorously researched could be reviewed and perhaps reused. In order to appreciate the suitability of approaches for use in knowledge sharing systems, access control models offered in the past needs revisiting. The inspection of access control mechanisms for file systems and databases are probably the closest one can get to.

The Bell-La Padula model[6] is one of the dominant early security models that are still being referred to till today. It is based on discretionary access control where access control is at the discretion of the object owner. Contrary to the discretionary access control model is the mandatory access control[6] where access rights are assigned centrally. The Biba model[6] focuses on guaranteeing integrity. Other dominant models that offers solutions to security issues included the Clark and Wilson, Harkness-Pitelli, Gorguen and Meseguer and Lipner as mentioned in[7,9] and discussed at length in[6].

Apart from earlier access control models, there exist many newer security frameworks which incorporated the earlier models mentioned. For instance, the Versatile Integrity and Security Environment (VISE) for Computer Systems Framework presented in[9] ensures confidentiality and integrity of objects governed by its framework. The framework offers a security shield for all applications under its care and exist as a separate layer above the existing computer's operating system. Although this framework may be effective for any applications registered under VISE to be protected, issues of deregistration and subsequent registration of all users and programs might arise if an existing computer fitted with this framework were to be given an operating system upgrade or change. As such this technique may be better off if it is developed as a part of an operating system. Knowledge sharing systems could thus enjoy the protection provided by this framework but it is our opinion that such systems must be capable of protecting themselves in the absence of such a framework.

Another implementation of security that may be considered to be partially applicable for knowledge sharing systems but is still tightly coupled to the operating system of the computer is the TRIPWIRE project as discussed in[7]. This is a file system integrity checker. TRIPWIRE was implemented to facilitate UNIX system administrators and users to monitor changes in a designated set of files or directories. The idea of file integrity checker is very useful in knowledge sharing systems to monitor its repository status.

Yet another possible adaptation of security for knowledge sharing systems can be seen in the implementation of the Coimbra project[5]. In this project, a modification of a web browser fitted with access control rules is introduced to enforce access control. The request to access a particular file will cause access control rules to be evaluated before they are presented to the user. Providing a mechanism to evaluate access rules is definitely beneficial in the case of knowledge sharing systems.

SECURE KNOWLEDGE SHARING AND RETRIEVAL

The approaches offered by earlier mentioned frameworks and models is observed to be somewhat applicable for knowledge sharing systems. As such, we took the approach of unifying the strengths of the isolated approaches into one integrated secure means for knowledge sharing and retrieval. Thus in[10], we introduced a Secure Knowledge Sharing and Retrieval framework (SKAR) for developing a knowledge sharing system. This approach models closely to the definition of knowledge sharing in[11,12]. It also conforms with the four major security requirements of usage auditing, user rights entitlements, encryption and repository access. In this framework, a typical knowledge sharing activity starts with the process of knowledge acquisition followed by knowledge retention and eventual retrieval for dissemination.

Initial access to the system is governed by the security manager which refers to user profiles and constraints for authentication. At each level of an activity, agents are deployed to check for consistency of access, constraints conformance as well as integrity of knowledge objects requested and submitted. Usage is constantly monitored and logged for auditing purposes. The SKAR framework is depicted in Fig. 1. Exchange of knowledge among members of a particular community becomes possible with the ability of users to freely decide on whether to share or not, what to share , whom to share with and how to share.

As illustrated in Fig. 1, the SKAR model comprises the interaction manager, security manager and the knowledge sharing engine. The interaction manager plays the role of providing users with navigation

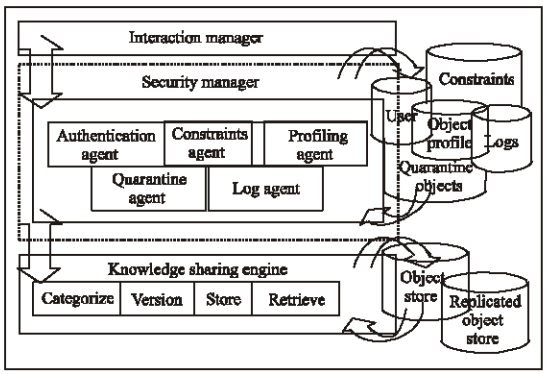

Fig. 1: The SKAR model. This model demonstrates the workings of a knowledge sharing system with security features

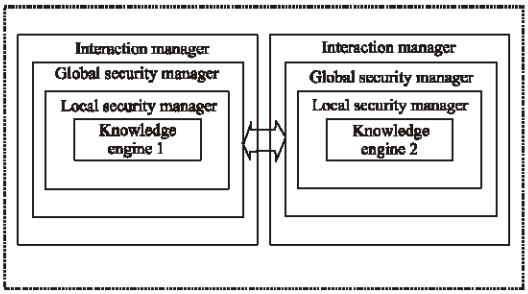

Fig. 2: Collaborative secure knowledge sharing and retrieval model (C-SKAR)

and presentation facilities. The security manager in turn ensures that all requests and responses to the user is guarded against a set of rules. To achieve this, the security manager engages a set of agents whose main tasks are to carry out the authentication, constraints checking, profiling, quarantining faulty objects as well as logging activities. Finally the knowledge sharing engine performs the specialized task of storing, retrieving, categorizing and versioning knowledge objects.

SKAR was designed to be very comprehensive in its features and its open architecture allows the a multitude of tailorable solutions in which individual knowledge sites can be different with regards to available interaction interfaces, archiving systems, internal work organization as well as technology such as server types and so on. However it was distinctly built to support a single secure knowledge sharing site. The security features in SKAR was not yet adequate to allow knowledge sites to collaborate with one another confidently. Rules that govern communication with external sites need to be stored, managed and enforced differently from those for local access. SKAR currently supports and manage knowledge objects and users that are registered to a single knowledge sharing site.

The advancement of data communications technology today allows different systems to interoperate independently. The emergence of Web Services[13] through the use of open standards like XML very much supports this kind of operation. Two or more knowledge sites can foster a relationship and allow a user who is a member of one site to access knowledge objects belonging to a different site without having the user to physically log on to the external site. Users may not be aware that he or she is actually accessing an external site.

Collaborative knowledge sites such as this would mean that users only need to remain a member of one knowledge site and yet enjoy the benefits of memberships to all other collaborative knowledge sites. With this new requirement, additional features must be introduced in SKAR to support such interactions. Since SKAR was designed using a layered approach, the additional requirement would mean minimal changes to the existing model.

In the next section we introduce an additional layer that fortifies SKAR and prepares it for a collaborative environment This expands the possibility of knowledge sharing systems to go beyond a single site and unleash the true meaning of knowledge sharing.

Collaborative secure knowledge sharing and retrieval: In this section, we introduce an additional layer above the SKAR's security layer to cope with online collaboration between two or more knowledge sharing systems. There will exist a global security manager in addition to the original SKAR security manager. This enhanced model termed C-SKAR will allow two or more participating knowledge sites to collaborate and share their knowledge objects with confidence.

Although the design of C-SKAR allows many participating knowledge sites to collaborate, the security restrictions imposed by individual sites will continue to be respected and enforced. For instance, a knowledge site can still decide whether or not to expose its knowledge objects to external sites as well as impose access conditions that apply differently to different sites. Fig. 2 demonstrates how a typical knowledge exchange can take place between two knowledge sharing systems securely.

Using C-SKAR , each site is equipped with an additional global security manager to enforce any restrictions during online collaboration with other sites. This is to protect the local knowledge sites from any

form of security violations during interaction with other requesting sites.

When a requesting site wishes to get knowledge objects from a local site, the Global Security Manager (GSM) of the local site is invoked. It will then resolve the request and inspects if the request has valid permissions. Following that it will check for any potential constraints violations. Once this has been ascertained, the request will then be subjected to local rules enforced by the Local Security Manager. If the requests are cleared to be free from any restrictions, the distilled request will then be attended to by the knowledge sharing engine and the resulting knowledge object handed to the requestor. At this stage, the GSM updates the list of objects that has been released to external requestors. The next subsection discusses the workings of the GSM in more detail.

The global security manager: The major enhancement to the original SKAR model is the introduction of the Global Security Manager. The workings of the Global Security Manager is very similar to the Local Security Manager except that the global security manager focuses on the security of interactions from requesting sites. Like the security manager in SKAR, the global security manager engages agents to complete its tasks.

Global authentication agent: The task of the authentication agent resolves the origin of a request and checks if the requestor has permissions to make the request. A corresponding list of permitted sites is referred for this purpose.

Global constraints agent: This agent's primary task is to check the requests against a global constraints list to ensure that the requests is legal. Constraints can range from simple constraints such as the number of downloads allowable to more complex one that involves content based constraints and aggregate constraints.

Release agent: This agent keeps track of previously released objects and updates objects released.

The GSM along with the other components of the original SKAR model now facilitates a secure interaction between other knowledge sites be it homogenous or heterogeneous.

GSM security enforcement: In this section, we focus on the methods used for global security enforcement in C-SKAR using GSM. From the earlier explanation on the GSM, it can be seen that the enforcement of global security is a consequence of a requestcoming from a remote site. Three main methods used by the GSM for enforcement of global security are: *authenticate_requests()*and *enforce_global_constraints()* and *release_results()*

When a remote knowledge site issues a request, the GSM from the local site will use the method *authenticate_requests()* to resolve the origin of the requestor and confirms if the requestor has the necessary permissions. The algorithm to authenticate the request is as follows:

Algorithm 1 (Site authentication): Input: Site_id and query Q
Output: Accept/reject the query
Method: The authentication is performed as follows:

- retrieve all information related to the requestor from the issued query Q: Requestor_label and request_labels
- retrieve the access control lists (i.e., access_control_lists) about participating sites
- Authenticate the requesting site as follows:
  If requestor_label exist in access_control_list
  Then requestor_label is authorized to make the request
  Else reject request of requestor_label

After authentication, the query is then translated and parsed to the global constraint agent for further security checks. The method *enforce_global_constraints()* is now invoked. The following algorithm summarizes the steps in enforcement of global constraints.

Algorithm 2 (Constraint checking): Input: given query Q
Output: verified query
Method: The enforcement of constraints is performed as follows:

- Invoke the global constraints agent to search for relevant constraints associated with query Q
- retrieve the associated constraints
- if there exists related constraints, modify the query according to the constraints. Otherwise if the query is not affected by the constraints, optimize and send out to Local Security Manager

At this stage, the GSM will invoke other methods to ensure that the query is properly transmitted to the Local Security Manager and to ensure the completeness of the transactions. We do not include these considerations in C-SKAR.

Once the results of the query is ready to be collected, the GSM will invoke the release agent to sanitize the results one last time before recording and presenting to the requesting site.

Algorithm 3 (Sanitization of query results): Input: A given query_result
Output: sanitized results
Method: The sanitization is performed as follows:

- retrieve the query_results from the Knowledge Engine via the Local Security Manager
- invoke the release agent to check for the knowledge objects that has been previously released to the requesting site
- invoke the global constraints agent to retrieve the relevant release constraints
- inspect the constraints and determine whether the result violates any of the constraints
- if the query violates the constraints, then prevent the results from being released to the user. Otherwise indicate the result can be released.

The cycle reach an end once the results have been post-processed and the results directed back to the requesting site. As pointed out, the algorithms allow the enforcement of security policies in C-SKAR.

C-SKAR STRENGTHS

Although the main issues of knowledge sharing systems are still to ensure proper authentication so as to be able to grant proper authorization and access control to users, this complexity has seemingly grown with the rapid evolution in data communication technologies. C-SKAR has the capability to handle both global and local security constraints. On the global scale, during interaction with other knowledge sharing systems, C-SKAR enforces access rules as well as constraints to protect the knowledge objects. At the same time, local access policies are enforced and the knowledge objects are maintained by using a failsafe redundant strategy that is able to detect any form of integrity violations to the knowledge objects and recover from it. Thus confidentiality, integrity and availability of knowledge objects are retained in addition to the flexibility of interacting with other sites.

CONCLUSION

As more knowledge sharing systems become increasingly interoperable, the related security issues become exceedingly complex. The security architecture that we presented is capable of protecting the knowledge acquisition and retrieval process through the enforcement of users authenticity, access entitlements, dynamic constraints and access rules imposed on objects in a collaborative environment. With the ability to glanularize the extent of sharing between collaborative sites, it is hoped that online collaborative knowledge sharing sites will become a commonality in time to come. Areas such as government services to people, entertainment, education and learning, health care etc are all potential beneficiaries from the emergence of collaborative knowledge sites on the Internet.

REFERENCES

1. Thorston, D., Robert and Sargent, 2002. Open Source initiatives for simulaton software: A Web-ready HiMASS: Facilitating collaborative, reusable and distributed modeling and execution of simulation models with XML, Proceedings of the 34th Conference on Winter Simulation: Exploring new fontiers, San Diego, California, USA, pp: 634-640.
2. Ernesto, D., D. Sabrina, P. Stefano and S. Pierangela, 2001. Fine Grained Access for SOAP E-Services, Proceedings of 10th International Conference on World Wide Web, Hong Kong, pp: 504-513.
3. Angi, V., T. Kreifelts and SOAP, 1997. Social agents providing people with useful information, Proceedings of the International ACM SIGGROUP Conference on Supporting Work group:the integration challenge, Phoenix, Arizona, USA, pp: 291-298.
4. Shaw, G., 2000. Digital document integrity, Proceedings of the 2000 ACM Workshops on Multimedia, Los Angeles, California, USA, pp: 143-144.
5. Weippl, E., 2000. Coimbra: Secure web access to Multimedia Content, Proceedings of the 2000 ACM workshops on Multimedia, Los Angeles, California, USA, pp: 145-148.
6. Fischer-Hubner, S., 2001. IT security and privacy, LNCS 1958, Springer-Verlag Berlin Heidel berg, pp: 35-106.
7. Harrington, A. and C. Jensen, 2003. Cryptographic access control in a distributed file system, Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies, Como, Italy, pp: 158-165.

8. Kim, G.H. and E.H. Spafford, 1994. The Design and Implementation of Tripwire: A file system integrity checker, Proceedings of the 2nd ACM Conference on Computer and Communications Security, Fairfaz, Virginia, USA, pp: 18-29.
9. Limoges, C.G., R.R. Nelson, J.H. Heimann and D.S. Becker, 1994. Versatile Integrity and Security Environment (VISE) for Computer Systems, Proceedings of the 1994 workshop on New Security Paradigms, Little Compton, Rhode Island, USA, pp: 109-118.
10. Ismail, A. and H.T. Ewe, 2004. Constraints Enforcement and Integrity Preservation for Knowledge Sharing Systems, Proc. 1st Intl. Conf. Telecommun. Computer Networks (IADAT-tcn 2004), San Sebastian, Spain, pp: 214-218.
11. Dignum, V., 2004. Personalized support for knowledge sharing, Proceedings of the Conference on Dutch Directions in HCI, Amsterdam, Holland, pp: 1-4.
12. Vishik, C. and A.B. Whinston, 1999. Knowledge sharing, Quality and Intermediation, ACM SIGSOFT Software Engineering Notes, Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration, 24: 157-166.
13. Chen, Li., Claus and Pahl, 2003. WWW applications: Security in the web services framework, Proceedings of the Ist International Symposium on Information and Communication Technologies, Dublin, Ireland, pp: 481-486.