CERIAS Tech Report 2002-07

Separating Between Trust and Access Control Policies: A necessity for Web Applications

M. Mahoui, B. Bhargava, Y. Zhong
Center for Education and Research in
Information Assurance and Security
&
Department of Computer Science, Purdue University
West Lafayette, IN 47907

Separating Between Trust and Access Control Policies: A necessity for Web Applications*

M. Mahoui, B. Bhargava, Y. Zhong

CERIAS and Department of Computer Science, Purdue University
West Lafayette, IN, 47907
{mmahoui, bb, zhong}@cs.purdue.edu

Abstract: As Security is the key of success for Web Applications most of the efforts that have been put in this domain have focused on wining users' trust to adopt the Web environment for their business operations. Although user trust is of paramount importance for Web applications, one also needs to consider Web applications trust towards users here after referred to as user trustworthiness. This paper explains why management of trust/mistrust is an increasing security issue in Web environment and proposes an authorization architecture framework that clearly separates between access control policies and mistrust management. It also describes a model that evaluates trustworthiness of users trust towards its integration in the authorization process.

1. Introduction

Recent advents in Web technology has driven more financial and other commercial transactions to take place online; and security is of paramount importance to succeed in Web environments [4]. As trust and trustworthiness are the foundations of security [6], gaining users trust has been by far the main concern of Web-based applications. Organizations and companies have focused on establishing more trust in online environments in order to bring more people to do business and to seek services over the Web. As described in [6] two classes of main security services are the basis for a secure internet infrastructure, access control services and communication security services. The latter insures the integrity and confidentiality of the information transmitted over the network and uses solutions such as SSL technology. The former ensures that information is not accessed or manipulated by unauthorized persons. Solutions used in this context include public key infrastructures (PKI) and more recently incorporating more complete solutions using access control models such as role based access control (RBAC) [3, 10]. These solutions although they provide confidentiality and integrity necessary for establishing users trust towards the Web servers, they don't address the problem of accountability of the users towards the Web servers. Unlike in closed systems such as operating systems, it is difficult to manage server trust towards the users. Strangers are continuously introduced into the system as long as they meet the criteria of the Web server trust establishment and recognized as authorized users; and the process of tracking their activities for accountability purposes is not straightforward as it is in traditional systems. This work emphasizes the importance of managing user trustworthiness, and proposes a new authorization model that clearly separates between access control policies and mistrust management. The recent incident reported in the press where two Lucent Technologies scientists have been accused of stealing Lucent software [11] emphasizes the importance of mistrust management in the architecture of authorizations systems in Web environments [12]. Mistrust management has an impact on security from two perspectives:

- Controlling users trustworthiness: unlike in current security systems in which user trustworthiness relies only on credentials (e.g. certificates, username/password), we propose to build user trustworthiness over time as the user interacts with the Web server. To take into account the open nature of the Web, functions of mistrust management should include the deployment of mechanisms for detecting external mistrust events that have an impact on the security of the

* This research is supported by the Center for Education and Research in Information Assurance and Security and NSF grants EIA-9805693, CCR-001788, and CCR-9901712. This paper is published in IEEE workshop on "Security in Distributed Data Warehousing", New Orleans, Oct. 2001.

system. Moreover, as the involvement of the users in externals mistrust events is not always possible to be clearly established, mistrust management should be able to characterize this uncertainty in evaluating user trustworthiness.

- Controlling Web resources declassification: the Web as a dissemination tool has motivated many companies to make available some of their online resources for external users. Examples include hospitals that want to share their large database of anonymous data for research purposes with other users such as external practitioners and certain hospitals [5]. In e-commerce, companies such as eBay have already started declassifying user profiles and made them available for authorized companies. The problem is that the process of declassification involves a large amount of information; and this increases the risk of declassifying non-intended data. For example, a declassified document may be utilized to infer sensitive information if it lands in the hands of "smart" external users. Security systems must include mechanisms for early detection of such anomalies to react accordingly.

The main contribution of this paper are as follows:

- Explain why trust management towards users, or simply mistrust management is important in Web applications.
- Propose a new architecture for modeling authorization over the Web that separates between access control policies and mistrust management.
- Propose a model for evaluating mistrust in the Web.

The rest of the paper is organized as follows. Section 2 describes the components of the new authorization architecture. Section 3 describes mistrust management and proposes a model for evaluating mistrust in the Web. Section 4 concludes the paper and points to some directions for future work.

2. Separating between Mistrust Management and Access Control

Efforts made in trust/mistrust management in open environments such as the Web have focused on trust establishment which is another variant of access control [2, 5, 7,]. Given a request to access an object, the system will decide whether the subject is allowed the access or not. Credentials presented by the subject are used to decide whether the subject is an authorized user or not using system's access control policies. Role based access control (RBAC) is receiving acceptance as the generalized approach for access control [3, 10]. On the other hand, credentials based on public key infrastructure (PKI) are considered as a solution for systems that need a strong authentication component compared to username/password based approaches [5]. In this new context, trust establishment is seen as the component that allows mapping of a set of credentials presented by a user to a role based access control. The evaluation of the credentials will determine the appropriate role that a user should have; and from that point on, the user is fully trusted as long as he/she accesses only the resources specified by the role he/she is playing.

We propose a new architecture that separates between mistrust management and access control policies such as RBAC (fig.1.a.). The objective is to facilitate the integration of the mistrust management component in existing security systems that use RBAC models. The major change that needs to be performed is to add a new level to the existing authorization system: given a request made by the user to access a resource, the system will decide whether to grant the request or not based not only on the role played by the user but also on the minimum trustworthiness that the user should meet to access the resource. Examples of authorization rules that can be generated with this new approach are the following: Request made by a user to access a Web resource is granted if:

- User has role A and his/her mistrust value is grater than 60%.
- User has role B and mistrust value equal to "fully trusted".

In addition, this approach will allow to easily extend previous work such as in [3, 5, 7] that integrates RBAC model in Web environments.

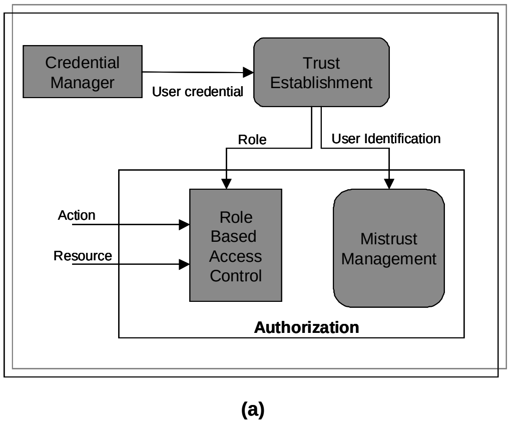

Fig.1.a. Mistrust Management with Role Based Access Control

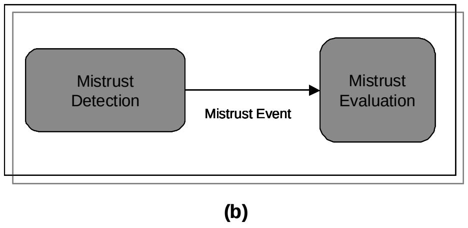

Fig.1.b. Mistrust Management Components

In the proposed architecture, the role of the credential manager is to collect enough credentials that allow the trust establishment component to identify the user and also to map the user identity to the appropriate role. Unlike in [5], our definition of credentials is not based on certificates only. Informally, we define a credential as the set of information collected about the user that is assessed to be sufficient enough to identify the user. Example of credentials include certificates such as public key certificates, username/password, or information collected from a survey questions presented to the user. The objective is to allow the architecture to adapt with several existing security systems that do not necessarily rely on certificates in the authentication process. The credential manager component has actually been studied as part of the work presented in [1, 9, 13] investigating static and dynamic security in Web environments. We intend to incorporate these results into the new architecture.

The focus of this paper is on mistrust management. Figure 1.b. identifies two main functions of mistrust management: *mistrust detection* and *mistrust evaluation*.

3. Mistrust Management

3.1. Mistrust Detection

Mistrust detection involves the deployment of detection techniques to capture mistrust events that have an impact on the security of the system. In Web environments, we are interested in detecting external mistrust events, and a particular type of mistrust detection for Web applications, that is concerned about information misusage. As the Web space continues to grow, one efficient method to narrow the external search space for mistrust events is to search for misusage of the information/services provided by the system.

More precisely, we use mining techniques for detecting information misusage. In [8] we propose a new architecture for information misusage detection called UserWatcher. This architecture integrates Web content mining and Web usage mining to monitor users access to Web servers. Discovered correlations between server documents and user documents can be used to detect information misusage. Information misusage may vary from (1) “straight plagiarism” to (2) the inference of information that was non-intended to be discovered by the user. In the first case, the system needs to revaluate user trustworthiness (see section 3.2); while in the second case the system needs to adjust its strategy for declassifying documents. UserWatcher uses a uniform representation for all Web resources (e.g. usage logs, user documents) to simplify the mining process. It relies on techniques such as similarity computation and clustering to compute correlations between documents.

3.2. Mistrust Evaluation

The problem of mistrust evaluation can be defined as follows: let us assume that the security system provides a mistrust detection mechanism such as UserWatcher. The detection mechanism could be either automatic or semi-automatic (if human involvement is necessary in the detection process). Given a mistrust event characterized by a set of parameters, and in which a user is involved, the question is how to use these parameters to update the current user's mistrust value. The evaluation of the new value will potentially depend on the previous mistrust value. Here, the involvement of the user could be either total if there is enough evidence that clearly identify the user as one of the actors in the mistrust event; or partial if the user is suspected in the event. The result is a new mistrust value assigned to the user which indicates how trustworthy the user is towards the system.

Formally we define user's mistrust by a function MISTRUST:

$$\begin{array}{lll} \text{MISTRUST} : U & \rightarrow & MT \\ \qquad user_i & \rightarrow & mistrust(user_i) \end{array}$$

Where U is the set of users and MT is the set of values representing the mistrust in a given system. In its simplest representation, MT values are numeric values representing percentages (e.g. 60%). For other systems ordinal values might be more appropriate (e.g. "fully trusted", "tolerated").

We propose a simple solution that aims to be as general as possible to be able to accommodate a variety of authorization systems using different mistrust detection mechanisms. The solution assumes that the domain of user mistrust (MT) is composed of a set of discrete values (e.g. "fully trusted", "tolerated"), and that the domain of each parameter that characterize a mistrust event is also composed of set of discrete values. Based these assumptions,, a finite state automata is used to model mistrust evaluation. The states correspond to the mistrust domain values. A transition from a state s_i to state s_j is based on the current state s_i and the values taken by each parameter that defines the mistrust event. We assume that the knowledge of the current state and the values taken by the mistrust event parameters are sufficient to determine the new state with 100% certainty. That is, no probability is associated when computing transition phases. A mistrust transition matrix is then provided to represent the finite state automata. As the computation of the transition matrix is not an automatic process, it is essential to:

a. Identify the minimum set of parameters that fully define a mistrust event in terms of its impact on mistrust evaluation.
b. Reduce the cardinality of MT set as well as the domain set of each parameter defining the mistrust event.

The assumptions that we made to construct the finite state automata address objective (b). Regarding objective (a) we identify three parameters that define the mistrust event:

- Data_sensitivity: Sensitivity of the data resource(s) affected by the mistrust event. Example of data_sensitivity domain is the standard classification of data sensitivity (top secret, secret, classified, etc.). In case more than one resource are involved in the mistrust event, a transition phase is generated for each resource. On the other hand, the coarse approach would be to assign a sensitivity value that is a combination of the sensitivity values associated to each resource affected by the mistrust event.
- Mistrust_scope: The percentage of initial users having access to the data who are involved in the mistrust event. For external mistrust events, it is not always possible to clearly establish the involvement of the exact persons in the mistrust event. Rather, the mistrust detection mechanism will provide a set of users that are "potentially" involved in the event. In other words, the mistrust action is spread over a set of users. Therefore, the number of suspected users when compared to the initial users set who had access to the resource has an impact on computing the new mistrust value of the suspected users. For example, if the mistrust event determines that one person among the initial 10 persons is suspected in the mistrust event, then updating the mistrust value of this user should not be processed the same way as if the mistrust detection mechanism identified the user among seven other persons suspected in the event.

- Involvement_uncertainty: This parameter captures the uncertainty inherent to the process of externals mistrust detection. Although we assume that the system will deliver a set of users potentially involved in the mistrust event, one should be able to define different levels of user involvement (e.g. “suspected”, “highly suspected”, etc.). The detection mechanism is either capable to assign appropriate degree of involvement for each user associated to the mistrust event or a single value will be associated to the set of suspected users.

In fact, the three parameters characterize the two components that compose a mistrust event: that is, the data affected by the mistrust event and the set of users involved in the event. Given a mistrust event characterized by its parameter values and the list of users suspected in the mistrust event, then the mistrust value of each suspected user will be revaluated using the appropriate entry in the mistrust transition matrix.

4. Conclusion

We have presented a new authorization architecture for securing Web resources that identifies mistrust management as a separate component from access control policies. Mistrust management ensures two roles: mistrust detection and mistrust evaluation. In mistrust detection we focused on information misusage detection using mining techniques. In mistrust evaluation, we define three parameters to characterizing mistrust events in terms of their impact in mistrust evaluation. We also propose a simple method for evaluating user trustworthiness, when user is potentially suspected in a detected mistrust event. We intend to conduct experiments to assess the relevance of the mistrust events parameters on evaluating user mistrust. In addition we intend to investigate new methods for mistrust evaluation based on machine learning techniques to allow the specification of a richer and more flexible domain for mistrust values as well as for mistrust event parameters.

5. References

1. B. Bhargava. Security in Data Warehousing, DaWak-2000, London, UK, 2000.
2. Y.H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, M. Strauss. REFEREE: trust Management for Web Applications, Word Wide Web Journal, pp. 127-139, 1997.
3. D.F. Ferraiolo, et all. A Role-Based Access Control Model and Reference Implementation Within a Corporate Internet, ACM Trans. Info. Syst. Security, 2(1):34-64, 1999.
4. S. Garfinkel, E. H. Spafford, Web Security and Commerce, O’Reilly and Associates, Sebastopol, CA, 1997.
5. Herzberg, Y. Mass, J. Mihaeli, D. Naor, Y. Ravid, Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers, 2000 IEEE Symposium on Security and Privacy, CA, 2000.
6. J.B.D. Joshi, W. Aref, A. Ghafoor, E.H. Spafford. Security Models for Web-Based Applications, communications of the ACM, 44(2):38-44, 2000.
7. N. Li, et al. A Logic-based Knowledge Representation for Authorization with Delegation, 12th Computer Security Foundations Workshop, 1999.
8. M. Mahoui, B. Bhargava, M. Mohania. Data Mining For Web Security: UserWatcher, IC’2001 conference. Also available as CERIAS TR 2001-20, Purdue University, 2001.
9. M. Mohania, V. Kumar, Y. Kambayashi, B. Bhargava. Secured Web access, Proc. of Kyoto International Conference on Digital Libraries: Research and Practice, 2000.
10. Proc. of the 5th ACM workshop on Role Based Access Control, Germany, 2000.
11. Reuters. Lucent scientists accused of system theft. http://news.cnet.com/news/0-1004-200-5811285.html, 2001.
12. J. Viega, T. Kohno, B. Potter. Trust and Mistrust in Secure Applications, Communications of the ACM, 44(2):31-36, 2000.
13. Y. Zhong, B. Bhargava. Authorization on Web Access, IC’2001 conference. Also available as CERIAS TR 2001-17, Purdue University, 2001.