Information Security Architecture-Context Aware Access Control Model for Educational Applications

N.DuraiPandian*, V.Shanmughaneethi & Dr.C.Chellappan*****

*Research Scholar, Computer Science Dept.,Anna University, Chennai – 25
**Lecturer, Computer Centre, NITTTR, Taramani, Chennai, India
***Professor, Computer Science Dept., Anna University, Chennai – 25, India

Abstract:

Information is an asset for every organization. The constantly increasing nature of computers systems on the functioning of organizations results in concerns about the threats to the information usage.

Security requirements approached at the organization level initiate the need for models that capture the organizational and distributed aspects of information usage. Such models have to express organization specific security policies and internal controls aiming to protect information against unauthorized access and modification and against usage of information for unintended purposes.

Existing systems follows Role Based Access Control models (RBAC) which are application dependent and whether they address the problems posed by mobile devices such as note books, Personal Digital Assistants (PDA), is an open question. What today's organization require is flexible, authentication on necessity, context aware access control and enforcement of dynamic authorization?

In this paper we propose Context Aware Information Security Architecture to fulfill the organization's security needs.

Keywords:
Access Control, RBAC, Information Security, Dynamic Authorization, Context aware Systems, and Authentication

1. Introduction

Many organizations are extending their operations to operate over internet. Education is also one such application. Almost every education system participants (e.g. colleges, Book stalls, Press, Insurance companies) have already implemented some type of computerized system to manage their operations. But they don't have any connectivity between themselves at all. A more integrated system will be a boon to the parents. Also it will be less costly for a parent to log into a common website to get the information of many colleges before admitting his ward into a college instead of getting into different web sites. A web services approach will do this but privacy and security of secret information should be ensured. To do this one should understand the security requirements of modern education system.

First education system needs a variety of authentication mechanisms instead of traditional password mechanism. Biometric and non-biometric methods are to be used. A practical online education system must accommodate a variety of authentication mechanisms. Secondly, even within a single college, there can be lot of applications that require making complex access control decisions If a student wants to see the marks of another student he cannot see it unless he is given permission by the faculty who is handling the subject. Traditional RBAC allows authorization based on roles but that is not sufficient to enforce policies that are dependent on run time parameters. Current education

applications require context aware information security architecture to enforce that.

Third the security requirements of education system need a very dynamic, flexible policy enforcement which should also handle unexpected situations. In the proposed approach authorization permission is either granted or rejected dynamically with the aid of centralized access control policy. Changing context is inferred & corresponding access privileges are assigned automatically. Context parameters are location, time, role, authentication trust level, type of information accessed.

Rest of paper is organized as follows. Section 2 discusses related works. Section 3 discusses the proposed context aware security Framework. Section 4 discusses the implementation of the proposed system. Section 5 discusses the proposed model with an application. Section 6 discusses conclusion.

2. Related works

Mandatory Access Control (MAC) is discussed [12][13] by T. Martzahn and K.J.Biba and Discretionary Access Control is introduced in [11] by Sandhu and Samarati. RBAC was introduced by Sandhu [1] where components of RBAC were discussed. Here authorization is given by assigning permission to roles than users. Bertino et al.[2] discusses Temporal RBAC which introduces time into access control architecture. Location & system status is introduced by Covington et al.[3] as constraints. In access control decisions, involvement of subject roles, object roles & environmental roles is incorporated by Moyer & Abamed in generalized RBAC [4]. Activities in a team are used as contextual information by Georgiadis & Mavridis [5] & Wang [6] in Team based Access Control Model. Context Sensitive by Kumar et. Al [7] is not applicable in distributed scenarios. Some more research issues in context related security applications are discussed by Neumann & Strembed [8] and Lee et. Al.

Applying RBAC model to applications distributed over internet is proposed by Taylor and Murty [9] & Joshi et Al [10] which discusses security model for authentication & access in distributed systems. But context is not integrated in this model.

In the proposed architecture context parameters are included to extend the RBAC model.

3. Context Aware Information Security Architecture

3.1 Authentication:

When a request is made to access information, first stage is always authentication. Biometric and non biometric – Digital techniques are used. Each technique is assigned with a trust level decided by the perceived reliability of the technique or given by the organization by their experience with different mechanisms. Trust level of retina is denoted as T (Retina). Generally T (Retina) > T (Iris) > T (Finger Print) > T (Password). To access highly confidential information a high trust level is needed. In case user is logged on to the system with lesser trust level, he will be asked to re login and request with a required trust level.

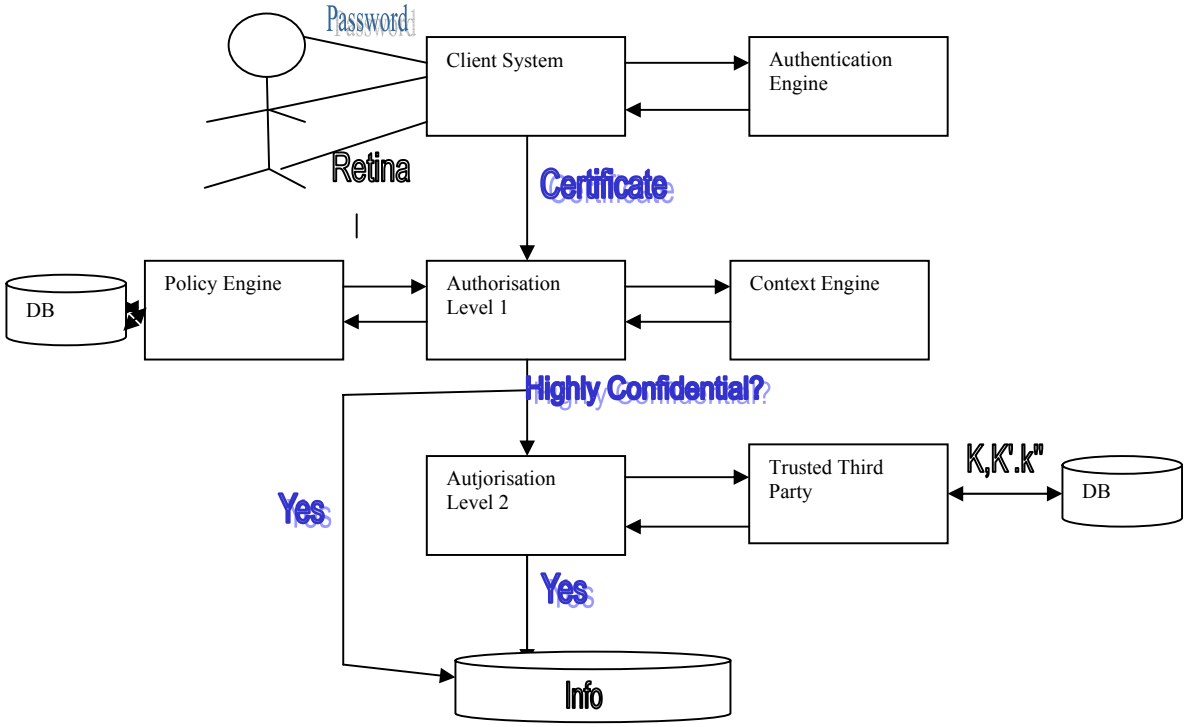

Fig.1 Security Architecture

3.2. Authorization Level 1

Education Systems have complex access rules since there are many actors in the system and their interlocking access privileges and most of the rules have to be context aware. An Educations system should support thousands of users, roles, objects and permissions. In this section context aware access control schema is introduced

3.2.1 Terminology Used:

Object: Smallest unit to be accessed and to be protected (ex.) Files

Object Set: Set of all objects within an application

User Set: Set of potential entities that can access objects in object set of an application (ex) Manager

Context Type (CT): Property related to every participant in an application where it is running.(ex.) Time

Context Set (CS): Set of Context Types (ex.) {Time, Location ….}. Whenever Necessary, elements of context set can be added by administrators.

Context Constraint (CC) It is expressed as a regular expression.

CC: =Clause1 Ụ Clause 2 Ụ …Clause$_i$

Clause: =Cond1∩ Cond2∩…Cond$_j$

Cond: = <CT> <OP> <VALUE> where CT Є CS, OP is a logical Operator from set {≤, ≥, ≠, ═, <,>}. we can have user defined operators also. VALUE is a specific value of CT.

3.2.3. Authorization Policy (AP):

It is a Quadruple.

AP = < UR, M. O, C>

UR is user or role

M is Mode of operation (Read, Delete, write and Update)

O – Object in question

C – Context Constraint

3.2.3 Object Access (OA)

It is also a quadruple.

OA = < U, M.O, DC>

U - User who made the request

M – Mode of Operation

O – Object
DC- Dynamic Context – Set of values for every context type in context set.
Object access is granted only if there exists an authorization policy <UR, M. O, C> such that UR ЄS, M=M. O=O, C evaluates to be true under DC.

3.2.4 Algorithm:

1) Find Policy Set required for a request
PSR = {}
For every AP in policy set
If (U in OA Є UR in AP) and
(M in OA = M in AP) and
(O in OA = O in AP)
Put AP in PSR
End if
End For
2) ACCESS = "Denied"
For every AP in PSR
New = C;
CL = TRUE;
For every clause CL in New
For every condition CN in CL
Get new value of CT in CN
Calculate CN with new value
If (CN=FALSE)
: CL=FALSE; Break ;}
End if
End For
IF (CL=TRUE) then Continue;
Else break;
End For;
If (CL=TRUE) ACCESS = "Accept"
Else ACCESS = "Denied"
3) If (ACCESS ="Accept") and (Classification= Highly Confidential)
Then use Ambient Calculus approach.

3.2.5 Ambient Calculus Approach

If the user wants highly confidential information then ambient concept is used. Assume user-agent trying to gain access to an ambient. In this case, we assume that the ambient, a firewall, keeps its name completely secret, thereby requiring authentication prior to entry. The agent crosses the firewall by means of previously arranged keys k, k', k''. The agent exhibits the key k' by using a wrapper ambient that has k' as its name. The firewall, which has a secret name w, sends out a pilot ambient k *[out w. in k'. in w]*, to guide the agent inside. The pilot ambient enters an agent by performing *in k'* and is given control by being opened. Then, *in w* transports the agent inside the firewall, where the key wrapper is discarded. The third name, k'' is needed to confine the contents of Q of the agent to prevent q from interfering with the protocol. The final effect is that the agent crosses the firewall and retrieves the information

Ambient Calculus Notation:

Highly confidential information P is kept in an ambient w. w is represented as a firewall.

Firewall ≅(def) (vw) w [k [out w. in k'. in w]| open k'. open k''. P].The request is represented by an agent that have three public keys k, k', k'' and the request as process Q**.**

Processes: P and Q
Ambients: w, k, k' and k''
Capabilities: in, out and open
Restriction: (v)
Firewall:
Firewall ≅(def) (vw) w [k [out w. in k'. in w]| open k'. open k''. P]
Agent:
Agent ≅ (def) k' [open k. k'' [Q]]
Agent and Firewall composition:
Agent | Firewall
≡ (vw) (k'[open k. k''[Q]] | w[k[**out w . in k'** . in w]| open k' . open k''. P]
→* (vw) (k' [**open k.** k'' [Q]| k [in w]]| w [open k'. open k''. P])
→* (vw) (k' [k'' [Q]| **in w**] | w [open k'. open k''. P])
→* (vw) (w [k' [k'' [Q]]| open **k'**. open k''. P])
→* (vw) (w [[k'' [Q]| **open k''**. P])
→* (vw) (w [[Q | P])
Here →* represents reflexive transitive closure and (vw) represents the restriction on w.

4. Implementation of Proposed Security Architecture

Fig. 1 shows the necessary system architecture of the proposed architecture.

Main Components of security architecture are

1) Authentication Engine
2) Authorization Engine
3) Classification Engine
4) Context Engine

Authentication Engine is responsible for issuing authentication certificate with trust level embolden on it. Authorization Engine monitors all requests coming from web service interface. If requester has right to access information which is decided at run time then access is granted and information is sent to requester. Context Engine evaluates each type dynamically and returns the results to authorization engine. Classification Engine provides the current classification level of object being accessed to authorization Engine.

5. Prototype of the system:

We discuss below an application "Education System" where the proposed architecture can be of more useful.

Application Overview:

The Education portal is the main access point of all users such as Professors, Students, and Administrative Staff etc. It provides information to common public and also provides dynamic function on web page for users based on their roles. For Example one user authenticated as a student can see his marks, assignment questions, test schedule etc. Also he can see the seminars being organized in the dept.,. When professor X logs in, all his classes, subject names are displayed. Selecting a particular one displays name of students enrolled, their previous semester Marks etc. This information will not be visible to others as they cannot authenticate themselves as Professor X. This is the first level of Security which is based on RBAC Model.

Behind this layer we have our second layer called context layer. Only system administrators can go up to this level to see those pages which define access policies, context definitions. This layer implements our dynamic context aware access control infrastructure & all access control policies are written there. They define context types, specify conditions associated with particular permissions based on context definitions.

This supports access control requirements that we need for education systems. One such requirement is a student can see his own marks but not other students. The administrator defines a context type isowner(Uid,Oid) which evaluates target object to determine if it is owned by target user and returns the Boolean value. To update student marks a professor should be the one who is handling the subject and he should be authenticated at a trust level of fingerprint or higher. This can be written as IsSubTaker(Uid,Sid) and AuthLevel(Uid)>=T(Fingerprint).

Performance Evaluation:

Let us see how the above architecture works in the following scenarios.

Following are some of the filenames available in the system.

{Colleges, Project, Career, Intercom numbers, Marks}

Object Type {Public, Confidential, Internal Use only, Highly Confidential}

Public: {Colleges, career}

Internal Use only={Intercom numbers}

Confidential = { Marks..}

Highly Confidential = {Project, Feedback about students}

Roles = { Management, Faculty, Student, Admin Staff, Public, Parents}
Users in Management =(Kumar, Vasu}
Faculty = (Kavitha, Kumaran….}
Students = { Eswar, James….}
Adminstaff={ Shanthi, Kala….}
Public = { Guest,….}
Parents = { Raman, Veean….}
Context Type = {Location, Time, Trust Level, Classification, Ismember(Project) IsOwner(User,Object)…..}
Location = {Secure, Public}
Time = {all time, office hours, non office hours}
Trust Level = {Password, Thumb, retina…}
Classification = {Public, Confidential, Internal Use only, Highly Confidential}
Action= {Low, Medium, High}
Low= {Read}
Medium= {Append, write}
High={delete}
Condition = <CT> <OP><VALUE>
Access Policy:
AP01:<Public, Read, Help, Location="public" ∩Time=alltime>
AP02:<Management, Read, Confidential, Location="Secure" ∩Time = "Office Hours∩IsOwner(User,Project) ∩Classifcation="Confidential"
AP03:<Student, Read, Marks, Location =Public∩Classification=Public> and we have so many other policies.
Scenario 1:
Guest requesting for college general information at 5.00 pm from a internet café
Req. info.: College = Public
Role = Public
Time = alltime
Location=Public

By our algorithm Rule AP01 will be selected.
There Condition is
Location="public" ∩Time=alltime
Location="public" ➔ *1*
Time=alltime ➔ 1
➔ 1∩1=1
So information will be given.
Scenario 2:
Kumar wants to see his project from internet café by 10.00a.m
Kumar is in management role. So AP02 will be selected. Conditions there is
Location="Secure" ∩Time = "Office Hours∩IsOwner(User,Project)
∩Classifcation="HighlyConfidential

Location = Public ➔ 0
Time = Officehours ➔ 1
IsOwner(Kumar,Project) ➔ 1
Classification ="HighluConfidential"➔1
===➔ 0∩1∩1∩1 ➔ 0
So he will not be given the access. Though he is having all rights but since he is trying to access the information from a public place, he is not authorized to see the information.
Likewise many cases are testes and found to be context aware. We have used XML for writing the policesbecause XML is the standard representation for interoperation rules between different applications.
.

6. Conclusion

In this paper we described a context aware information security architecture that extends the traditional RBAC model to gain more advantages because of the context property. Our research motivation comes from the complicated access control requirement of current education systeM,. Traditional RBAC model is static with poor flexibility and extensibility. Our new security infrastructure is dynamic and with following advantages.
Our proposed model takes authorization decisions based on context information
in addition to roles. It can be applied dynamically.
The proposed security infrastructure is flexible and allows easy extensibility.

References

[1] Ravi S Sandhu, Edward J.Coyne, Hal L Feinstein, Charles E.Youman, "Role Based Access Control", IEEE Computer, Volume 2, Feb. 1996, pp 38-47.

[2] Elisa Bertino Piero Andrea Bonatte & Elena Ferrari, " TRBAC: A Temporal Role Based Access Control Model,ACM Transactions on Information & System Security, Volume 4, No. 3, August 2001, pp 191-223

[3] Michael J Covington, Wende Long & Srividhya Srinivasan, " Secure Context Aware Applications using Environmental Roles", Proceedings of sixth ACM Symposium on Access Control Models and Technologies, Msy 2001, Virginia, USA

[4] M.J.Moyer & M Ahamed, " Generalised Role Based Access Control:, 21st International Conference on Distributed Computing Systems", April 16-19, 2001, Atlanta, USA

[5] Christos K. Georgiadis, George Pangalos, Ioannus Marvidis Rashan K. Thomas, " Flexible Team Based Access Controls using Contexts, Proceedings of sixth ACM Symposium on Access Control Models and Technologies, May 2001, Virginia, USA

[6]Weigang Wang. "Team & Role Based Organization Context and Access Control for Co-Operative Hypermedia Environments", Proceedings of tenth ACM Conference on Hypertext & Hypermedia, Feb. 1999, Germany

[7] ArunKumar, Neeran Karnik & Girish Chafle, "Context Sensitivity in RBAC", ACM SIGOPS Operating Systems Review, Volume 36, Issue 3, July 2002

[8] Gustaf Neuman & Mark Strembeck, "An approach to Engineer & Enforce Context Constraints in an RBAC Environment:, Proceedings of Eighth ACM Symposium on Access Control Models and Technologies, June 2003, Italy.

[9] Kerry Taylor, James Murty, "Implementing RBAC for Federated Information Systems on the Web", Proceedings of the Australassian Information Security Workshop, January 2003, Adelaide, Australia.

[10] James B.D.Joshi, Walid G.Aref, Arif Gnafor & Eugene H.Spafford. "Security Models for Web Based Applications", Communications of ACM, Volume 44, No.2, February 2001, pp 38-72.

[11] R.S.Sandhu , P.Samarati, Access Control: Principles & practice, IEEE communications 32(9), page 40-48, 1994

[12] K.J.Biba, Integrity Constraints for secure computer systems, EST TR-76-372, ESD/AFSC, Hanscom AFB, Bedford, MA, April 1977

[13] T. Martzahn, Implementing the Just Enough Privilege security model, SANSInstitute,Information security room, http://www.sans.org/rr/papers/56/1256.pdf

About the Authors:

Dr.C.Chellappan received his Master of Engineering Degree in Computer Science & Engineering and Ph.D. Degree in the area of Data Base Technology in the year 1987 from Anna University, Chennai, Tamil Nadu, India. He is currently working as Professor, Department of Computer Science and Engineering, Anna University. His Research interests include Computer Networks, Mobile Computing, and Network Security. He is guiding many Research Scholars in these areas from M.S. and Ph.D. programme. He has published many Research papers at National and International level Conferences and Journals.

Mr.N.DuraiPandian received his Bachelor of Engineering Degree in Electronics and Communication Engineering in 1989 and Master of Engineering Degree in Computer Science and Engineering in 1994 from Regional Engineering College, Trichy, TamilNadu, India. He is currently working as Professor, Department of Information Technology, Velammal Engineering College, Chennai, Tamil Nadu, India. His Research interests include Operating systems, Computer Networks, Network Security.

Mr.V.Shanmughaneethi received his Bachelor of Engineering Degree in Computer Science and Engineering in 1996 from Shanmugha College of Engineering, TamilNadu and Master of Engineering Degree in Computer Science and Engineering in 2002 from Anna University, Chennai, TamilNadu, India. He is currently working as Lecturer, National Institute of Technical Teachers Training and Research (NITTTR), [Govt. Of India] , Chennai, Tamilnadu His Research interests include Computer Networks, Network Security, and ICT in Education.