THE 15TH EDITION OF THE INTERNATIONAL CONFERENCE
EUROPEAN INTEGRATION
REALITIES AND PERSPECTIVES

Comparative Study of Access Control Methods in Enterprise Information Systems, Based on RBAC, ABAC, and TBAC policies

Marcel Danilescu[1]

Abstract: Controlling access to a company's IT systems is a way to ensure that users are the ones who say they are and have proper access to company data and documents. At a high level, controlling access to a company's data and applications is a selective restriction on access to data. It consists of two main components: authentication and authorization. Authentication is used to confirm that someone is the claimant, and this is not enough for themselves to ensure data protection. Authorization is additional levels which determines which user should be allowed access to data or perform an action (operation / transaction). For their implementation, several authentication and authorization methods have been created, of which, within this study, we approach, Role Based Access Control (RBAC), Attribute-based access control (ABAC) and Trust-based access control (TBAC). This study makes a comparative analysis on the principles underlying RBAC (Role Based Access Control), ABAC (Attribute-based access control) and TBAC (Trust-based access control) and the ways of application and collaboration between them.

Keywords: Users; operations; actions; objects; roles; trust; attribute

Introduction

The cyber threat nowadays is very high, and organizations (public or private) have to deal with possible external or sometimes internal attacks. They often have a devastating effect that can lead to a loss of reputation and potential business partners.

Organizations seeking to regulate the management of access rights in accordance with internal management policies face complex and time-consuming management for thousands of users, and difficulties in applying business-level control of access rights, which means management constraints on IT resources.

To meet these challenges, several methods have been used to design and implement authentication and authorization policies such as: Role Based Access Control (RBAC), Attribute-based access control (ABAC), Trust-based access control (TBAC).

[1] PhD in progress, "Danubius" University Galati, Romania, Address: 3 Galati Blvd., 800654 Galati, Romania, Corresponding author: marcel.danilescu@aswic.ro.

1. Role Based Access Control - RBAC

The concept of role-based access control (RBAC) (David F. Ferraiolo, D. Richard Kuhn, 1992) (Ravi Sandhu , David Ferraiolo , Richard Kuhn, 2000) has its foundations in the multi-user and multi-application systems of the 1970s. The main idea of RBAC is that the permissions granted to perform various actions/operations are associated with the various roles they have within the organization, and users are assigned the appropriate roles. This leads to a simplification in the management of permissions. Roles correspond to the various functions in organizations, and users are assigned roles according to their responsibilities and qualifications. Users are reassigned to various roles. Roles may be granted new permissions as new applications or systems are incorporated, or various permissions may be revoked for certain roles, as needed.

A role is a semantic construction around which an access control policy is formulated. The concept of role was created because the activities or functions of an organization usually change less, which leads to a certain stability.

A role may be the ability to perform certain specific tasks, such as a doctor or pharmacist.

A role can represent authority and responsibility, for example, the supervisor of a project. Authority and responsibility are distinct from competence. An employee may be competent to lead several departments, but is assigned to lead one of them. Roles can reject specific task assignments that are can be assigned to multiple users, for example, a doctor on duty or a shift manager. RBAC models and implementations should be able to conveniently adapt all these properties of the role concept (Ravi S. Sandhu, Edward J. Coynek, Hal L. Feinsteink, Charles E.Youmank, 1996). (Role-Based Access Control Models - Ravi S. Sandhu, Edward J. Coynek, Hal L. Feinsteink and Charles E. Youmank IEEE Computer, Volume 29, Number 2, February 1996, pages 38-47. Revised October 26, 1995):

The RBAC model is organized in 4 levels of functionality:

- Flat RBAC
- Hierarchical RBAC
  - general hierarchical RBAC
  - hierarchically restricted RBAC
- RBAC constrained
- Symmetrical RBAC

RBAC sets permissions based on functional roles in the enterprise, and users are assigned their roles and sets of roles.

Each level includes the previous one and in addition brings new requirements. (NIST Model For Role Based Access Control Towards A Unified Standard: Ravi Sadhu, David Ferraiolo, Richard Kuhn.

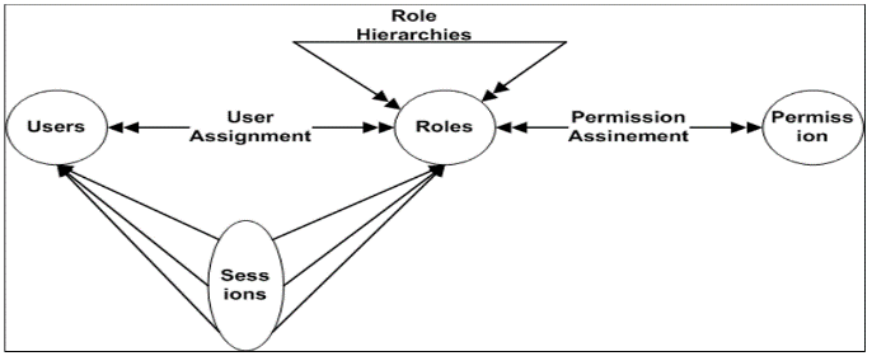

Figure 1. RBAC Model

The RBAC model has the following components:

U, R, P and S (users, roles, permissions and sessions), respectively

PA⊆ PxR, a multi-to-many relationship between permissions and roles,

UA ⊆ UxR, a many to many relationships to assign a role to a user.

Basically a user is assigned one or more roles, and a role is assigned one or more permissions. Multiple users can play the same role. Roles can have a hierarchy within the organization. During a work session a user is given a role that has certain permissions and may suffer certain constraints.

Implementing an access control solution involves analyzing user activity for a workstation and a specific task.

2. Attribute-Based Access Control –ABAC

In 2014, NIST published the "Guide to Attribute-Based Access Control (ABAC) Definition and Considerations" (Vincent Hu ,David Ferraiolo , Richard Kuhn , Adam Schnitzer,Kenneth Sandlin , Robert Miller,Karen Scarfone , 2014) (David Ferraiolo, Ramaswamy Chandramouli, Vincent Hu,Rick Kuhn, 2016).

The new standard replaces RBAC, and implies an increase in access control in computer systems. Attribute-Based Access Control (ABAC): an access control method in which the subject requests to perform operations on objects is granted or denied based on the subject's assigned attributes, object-assigned attributes, environmental conditions, and a set of policies specified in those attributes and conditions

Attribute-based access control (ABAC) is a method of access control in which the subject requests to perform operations on objects is granted or denied based on the subject's assigned attributes, object-assigned attributes, environmental conditions, and a set of policies specified in those attributes and conditions

ABAC has as main components:

Attribute - is a feature of any element in the network. An attribute can define:

- A characteristic of the user - his position, job, IP address, tasks, etc.

- A characteristic of the object - type, sensitivity, necessary level of training, etc.
- A type of action - reading, writing, editing, copying, etc.
- An environmental characteristic - time, day of the week, location, etc.

• Subject - any user or resource that can perform actions on the network; attributes are assigned to a subject to define its level of action;

• Object - any type of data stored on the network; objects are assigned attributes to describe and identify them;

• Operation - any action taken by any subject in the network;

• Policy - a set of rules that allow or restrict any action. Data processing; rules are "IF / THEN" statements based on attributes of any element (user, resource, environment).

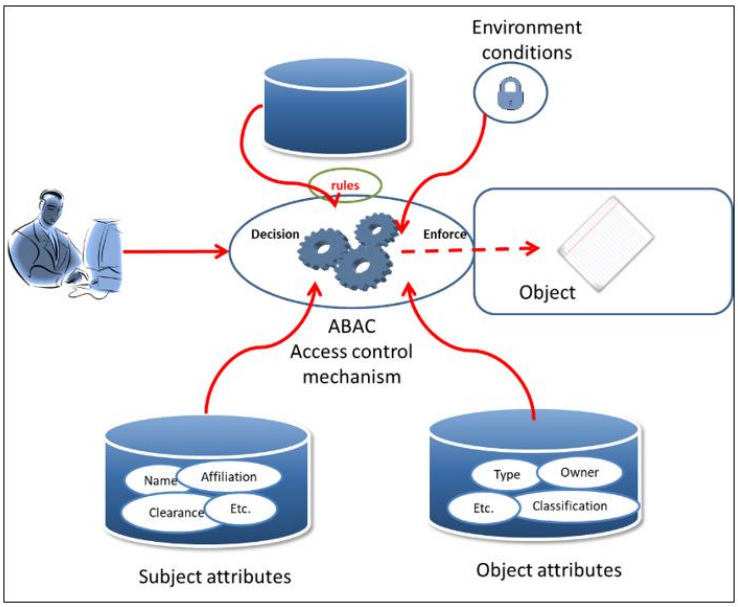

Figure 1. ABAC Baseline Scenario

Unlike RBAC, in ABAC you can use even attributes that are not yet registered in the system, but which will appear during the work process.

Four components are important in the implementation of ABAC:

• Policy Enforcement Point (PEP), responsible for protecting the application;

• Policy Decision Point (PDP), responsible for processing the application received and evaluating it in accordance with the authorization policies with which it was configured;

• The Policy Information Point (PIP) is the driver that connects the PDP to the underlying attribute sources;

• The Policy Administration Point (PAP) and is the tool by which administrators create, manage and edit authorization policies.

One of the key advantages of ABAC is that it standardizes how to query the authorization. Unlike other existing frameworks (spring, or claim-based, etc.) that have different ways of querying their own authorization engines, the ABAC authorization approach comes down to a simple "Yes" or "No" answer. This applies no matter how complicated the question is. The

simplicity of the request / response process facilitates integration with various applications and frameworks.

One of the key advantages of ABAC is that it standardizes how to query the authorization. Unlike other existing frameworks (Spring, or claim-based, etc.) that have different ways of querying their own authorization engines, the ABAC authorization approach comes down to a simple "Yes" or "No" answer. This applies no matter how complicated the question is. The simplicity of the request / response process facilitates integration with various applications and frameworks.

ABAC also allows you to have richer response flows. Although the answer is always in terms of answering a Yes / No question, the decision can be augmented with additional statements, such as:

- Yes, allow + record that the user has received access. Or
- No, deny + redirect the user to a two-factor authentication page.

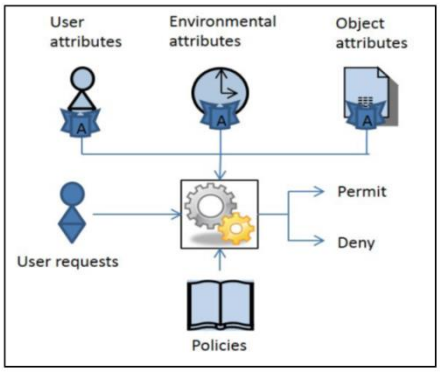

Figure 2. ABAC Policies

ABAC policies can be applied together with RBAC policies, increasing the degree of user access control.

3. Trust-Based Access Control – TBAC

Trust-based access control policies were researched and presented in 2010 (Danilescu & Danilescu, 2010) (Danilescu & Danilescu, 2010). In 2012, the paper "Data security management applying trust policies for small organizations, ad hoc organizations and virtual organizations" was presented (Danilescu M. , 2012).

Their purpose was to ensure access control in small, medium, ad hoc and virtual organizations.

A trust policy has been defined as follows.

Let be $O_i \in GO \wedge P_i \in \boldsymbol{P}$ *where* $P_i=(p_1,p_2,\dots p_k \dots p_n)$ *, and* $p_k=H_k(A_k)\ H_k(E_k)\ F_k$

for $\forall A_k$, $\exists(U_k \in G_m \Rightarrow \exists R_u$, $R_u(U_k)=R_a(A_k) \wedge R_u(U_k)=\leq R_g) \oplus$

$\exists(U_x \in G_m \Rightarrow \exists R_u$, $R_u(U_x)=R_u(U_k) \wedge \exists de_v(U_k)$ *for* $U_x) \oplus$

$\exists(U_x \in G_m \Rightarrow \exists R_u,\ R_u(U_x)=R_u(U_k) \wedge \exists de_v(U_k)$ *for* $U_x \Rightarrow re_v(U_k) \in RE$

$\wedge \neg \exists re_v(Ux) \in RE\ )$

Where:

A_k= an action applied to one object;

de_v= delegation received from a user U_k;

DE= the crowd of delegations;

F_k= flow sequences;

G_m= User group of which one user U_k is part;

GO=Group of objects;

$H_k(A_k)$= the corresponding action hierarchy to the p_k subprocess;

$H_k(Ek)$= the corresponding hierarchy of events to the *pk* subprocess;

O_i=Object *i*;

P_i=The process applied to O_i;

P=the set of processes

$p_1..p_n$ =numbers of *subprocess ale* P_i;

R_u= confidence level of the *U* user , that is needed for the *Oi* object;

R_g= confidence level for the *GM* group;

$R_a(A_k)$ = level of confidence necessary to the enforcement of the *Ak* action;

re_v=restriction applied to the user U_k;

RE= the crowd of restrictions;

U_k= the user designed to execute the A_k action;

U_x = an user which belong to the group G_m.

Applications generally follow a workflow, therefore, to implement the trust-based solution, the workflow must be designed, users established, the objects they interact with, their actions established and access control policies written.

An access control policy is in the form of:

{Oi,Uk,(Ak,rev,dev) }

The application of trust policies is done from the design phase, the workflow being determined and divided into categories of users. For small, medium, virtual and ad-hoc enterprises where the number of applications is small, and the number of users also allows the rapid creation of access control and user actions.

Conclusions

In this paper we presented three different concepts of user access control to the resources of an information system.

Each of the three concepts is well defined and acts differently on information system resources.

RBAC allows the creation of roles and their assignment to various users during work sessions, in order to have access to various resources and undertakes operations on them.

ABAC controls various attributes of subjects (which can be users, operating system tasks, etc.), resources, and depending on them allows / rejects the execution of certain operations by subjects on resources.

TBAC allows users, depending on the trust given for the execution of a certain action, to perform it on the data, documents within an information system.

Hybrid implementations were also made between ABAC and RBAC (Richard Kuhn , Edward Coyne , Timothy Weil, 2010), (Qasim Mahmood Rajpoot, Christian Damsgaard Jensen and Ram Krishnan, 2015). And TBAC can be seen as an ABAC that has an additional attribute of trust for both the user and the object.

References

Danilescu, Laura & Danilescu, Marcel. (2010). Control Access to Information by Applying Policies Based on Trust Hierarchies. *International Conference on Computer and Software Modeling, ICCSM 2010*, pp. 285-290. Manila: Institute of Electrical and Electronics Engineers, Inc.

Danilescu, Laura & Danilescu, Marcel. (2010). Organization's Data Access Control Policies Based On Trust. *Euroeconomica. 2*, pp. 113-122. Galati: Universitatea Danubius.

Danilescu, M. (2012). Data Security Management Applying Trust Policies for Small Organizations, Ad Hoc Organizations and Virtual Organizations. (D. Jurnals, Ed.) *The Journal of Accounting and Management*, 2(3), pp. 47-64.

Ferraiolo D. & Richard Kuhn. (1992). Role-Based Access Controls. *15th National Computer Security Conference*, pp. 554-563. Baltimore Md: National Institute of Standards and Technology/National Computer Security Center. Retrieved from Https://Csrc.Nist.Gov/CSRC/Media/Publications/Conference-Paper/1992/10/13/Proceedings-15th-National-Computer-Security-Conference-1992/Documents/1992-15th-NCSC-Proceedings-Vol-2.Pdf.

Ferraiolo David; Ramaswamy Chandramouli; Vincent Hu & Rick Kuhn. (2016). *A Comparison of Attribute Based Access Control (ABAC) Standards For Data Serviceapplications.* Gaithersburg, MD: NIST Special Publication. Doi:10.6028/NIST.SP.800-178.

Qasim Mahmood Rajpoot; Christian Damsgaard Jensen & Ram Krishnan. (2015). Integrating Attributes into Role-Based Access Control. *Proceedings Of The 29th Annual IFIP WG 11.3 Working Conference On Data And Applications Security Andprivacy*, pp. 242-249. Fairfax, VA, USA: Springer Verlag. Doi:10.1007/978-3-319-20810-7_17.

Ravi S. Sandhu; Edward J. Coynek; Hal L. Feinsteink & Charles E. Youmank. (26 October, 1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), pp. 38-47. Doi:10.1109/2.485845.

Sandhu Ravi; Ferraiolo David & Kuhn Richard. (2000). The NIST Model For Role-Based Access Control: Towards A Unified Standard. A. F. Machinery (Ed.), *RBAC '00: Proceedings Of The Fifth ACM Workshop On Role-Based Access Control*, pp. 47–63. Berlin: Association For Computing Machinery, New Yorknyunited States. Retrieved From Https://Doi.Org/10.1145/344287.344301.

Richard Kuhn; Edward Coyne & Timothy Weil. (2010). Adding Attributes To Role-Based Access Control. *Computer (IEEE Computer)*, 43(6), pp. 49-71. Doi: DOI: 10.1109/MC.2010.155.

Hu Vincent; Ferraiolo David; Kuhn Richard; Schnitzer Adam; Sandlin Kenneth; Miller Robert & Scarfone Karen. (January, 2014). *Guide To Attribute Based Access Control (ABAC) Definition And Considerations.* Retrieved 05 26, 2019, From Computer Security Resource Center: Https://Csrc.Nist.Gov/Publications/Detail/Sp/800-162/Final.