Sample Paper for Poster 2015 Conference for LaTeX

First Student AUTHOR[1], Second Student AUTHOR[2]

[1]Dept. of Radioelectronics, Czech Technical University, Technická 2, 166 27 Praha, Czech Republic
[2]Dept. of Physics, Another University, Another Street 123, Another City, Another Country

First.author@fel.cvut.cz, secauthor@supermail.com

Abstract. *The abstract of the paper brings very brief information about the contents of the paper. The abstract should not be shorter than 80 words and should not exceed 200 words. For the text of the abstract use the environment* `abstract`.

For typing the paper in LaTeX use the documentclass `poster15`*, the basic commands and environments are redefined according to the specific conference format.*

For the title of the paper use the command `\title{}`*. The name, the affiliation, and the e-mail address of the author(s) are of the styles* `author{}`*,* `\affiliation{}` *and* `\email{}`*. The affiliation is required being specified as shown in this example. All authors must have a student status!* ***Failure to comply with this rule may lead to rejection of the paper****.*

Keywords

Paper formatting, Poster2015, electronic publishing, templates, styles.

1. Introduction

Currently there can be seen move toward context aware (CA) application. It is caused by emerge of the huge amount of the mobile technologies and users demand for personalized applications. Applications provide personalized context based on user's context or the application's context. That brings completely new experience for the applications operators as well as for users. However securing the applications is done the old way. Usually users are assigned various roles in applications or permissions for resources and those security rules are independent on context. There is very few, if any, applications, which has security based on context. We can expect that users and application owners would take advantage of application security based on context to provide specific security rules based on users context.

Applications using CA security can be much less obtrusive for users. They can be asked for different authentication methods based on context, they can be authorized for same resource various ways depending on their context. They can even sometimes omit authentication because their context is trustworthy by itself (for example access from inner company network). Same as the users can profit from the context based authentication operators of the applications. They might define more strict security rules for suspicious users behavior (for example access to confidential resources in system from internet in night). Using context allows system administrators for more fine grained security rules, which would be otherwise unsustainable for maintenance.

Application operators and software developers are good aware of the added value of CA security. Even there are various proposals how to do CA security none of them is widely used. Reason why they are not widely used can be that they are either too complicated or they are too innovative and it is hard to incorporate them into existing solutions.

In this paper I will present solution, which extends standard role based security architecture with CA elements. This extension is based on giving users security level based on their context. Resources would require user to posses that level in addition to his normal rights to be accessed. That way we can extend any existing security architecture with CA elements.

2. Background

Large applications or information systems, which has more then one user needs some form of authentication and authorization. Such systems are here for many decades and are almost as old as computers itself. Therefore for many decades there exist problem with security applications and it was solved various ways.

Two of the oldest principles for securing application resources is mandatory access control (MAC) and discretionary access control (DAC). Those two principles does not define how the application security should be implemented, but rather define core principles in authorization. In MAC there is some authority, which grants permissions to all re-

sources. In DAC on the other side can grant permission everyone with sufficient permission for the resource.

However granting permissions to every user in system is unpractical for larger amount of users. Role based access control (RBAC) provides another level of abstraction. It that approach the permission is given to an abstract role and users are assigned roles. Typically there is in application many times less roles then permissions and the roles does not change significantly over time unlike users.

Nevertheless those authorization principles and method are very static. They does not reflect the changing state of the system and users. Once they are set they do not take in account any other factors and any fine tuning is if not impossible very difficult.

CA security can overcome those difficulties and provide even new experience for user and application operator. CA applications are much more personalized then the static ones and same comes for the security. Application can get a lot of information about user from context and therefore it does not annoy user with request for additional information. For example application knows his IP address and therefore does not need to re-verify him. The context is also valuable source of information for application owner. For example he might restrict some IP ranges, days of time, etc to access resources in application.

Also with the emerge of the context aware application there is naturally need of CA security. ADD DATA ABOUT CONTEXT AWARE APPs ADD DATA ABOUT CONTEXT AWARE APPs ADD DATA ABOUT CONTEXT AWARE APPs ADD DATA ABOUT CONTEXT AWARE APPs ADD DATA ABOUT CONTEXT AWARE APPs ADD DATA ABOUT CONTEXT AWARE APPs ADD DATA ABOUT CONTEXT AWARE APPs ADD DATA ABOUT CONTEXT AWARE APPs

To illustrate how can context aware security improve applications consider following example. We have a information system in company. To make it more comfortable let users from inner company network access some noncritical resources. But if the user comes from internet or access some sensitive resources he needs to authenticate itself normal way. But not only users would benefit from it. CA security can determine suspicious users behavior. For example if users log in to system in short time frame from different parts of world it can rise flag and the incident can be investigated further. Or the company can set some access hours for various resources, to limit possibility of their abuse (for example limit access in non working hours).

3. Related Work

There has been multiple attempts to extend classic role based access control with context-aware elements and make RBAC more fine-grained.

One of the approaches is to add another set of roles to role based access control. Moyer [1] proposes creating two additional sets of object roles and environmental roles and tying permissions with trio of roles. Covington [2] simplify that to just one additional set of environmental roles. They are hierarchical composed and represent current state of system. Similarly to this approach Seon-Ho [3] suggests additional set of context roles.

Different solution proposes Sladić [4]. Roles are granted to user after his authentication based on context. That way user can obtain new roles based on context. The idea is further developed by Kulkarni [5] into Context-Aware RBAC. It also allows roles to be granted based on context but there is second layer of authorization architecture, which is responsible for granting and revoking roles when the context changes and therefore roles are dynamically reflecting context.

There is also possibility to solve that problem with adding another element not based on roles. Neumann [6] suggests adding context constrains to security policies. When the permission is checked used needs to posses not only the permission for resource (based on his role) but also fulfill context constrains. Similar approach by Mostéfaoui [7] proposes that security rules should consist of four elements - permission, role, context and authentication method.

Lima [9] adds another context dimension to current security rules. It would make security policy three dimensional with context, permission and role. Difference from xoRBAC [6] is that it takes context more abstractly and complexly. Corrad [8] suggest leave the roles completely and assign permissions to contexts. Both those approaches are interesting in that they somehow compare contexts and make decisions on how similar they are.

Remarkable idea is proposed by Hung [10]. He proposes three entities in security rules - object, user and activity. All of the entities have some credentials. If user want to perform action on object he needs to poses credentials required for both object and activity.

Another interesting idea is described by Wendong [11]. He suggest adding user security level in addition to RBAC and define needed security levels to perform actions.

4. Proposed Solution

Security policies in organizations are very consistent and are changing just slightly over the time. Most of the organizations does not want and even need any radical change. Therefore CA security must be another logical step to evolve current security. This will allow us to build new security rules on existing and well proved solution.

I propose create security level, which is based on context in addition to traditional roles in RBAC. Level can be understood as it quantifies how is the user trustworthy and it

is dynamically tied to user. The security level creates second security constrain beside traditional permission and therefore resources in application now can have two different kind of security rules - classic permission tied with role and security level.

As the context of the user and the application is changing the level needs to reflect that dynamic nature of context. There is several moments when the level can be calculated. First is to calculate level during user's account creation. However this does not reflect dynamic nature of context and therefore is unsuitable. Opposite extreme is to determine level on every security check. This would reflect changing context most reliably but it is too demanding for computational resources and time consuming as the context check might not be trivial. As the best compromise seems to determine level during user's log in into application. It decreases number of context check by several orders and in the same time it provides very accurate snapshot of the user's context. In case the context would be changing rapidly the user can perform relogin or even the application can force new level calculation manually.

Resolving level is done by context resolvers. Each resolver takes responsibility for checking one particular part of context. For example one resolver would determine network, which user came from. Another would check time of the day and so on. Every resolver would return, which level it grants to user. The level does not need to be set in resolver and it decide just if to give it or not, the resolver itself can decide, which level to grant. After every resolver performs its inner logic and determines level on its own the highest level is used as the final user's security level.

Level representation by itself is very abstract. Only need for level is to be comparable with other levels to know if given level is higher or lower then required one and also to determine the highest one. Therefore it is not important whether level is represented by number, string or even some more complex structure. This leaves a lot of space for customization for given application.

The proposed solution has many advantages. The most important ones are:

- Lightweight - it does not require any complex structures in application nor it does not consume significant system resources.
- Easy to use - it just requires adding another type of constrain to resources that need to poses CA security.
- Voluntary - if someone wants to use plain RBAC he can and just to chosen resources he might add level restrictions.
- Scalable - there is not any predefined set of levels nor there is no limit in amount of levels in application.
- Universal - the solution can be modified and used with other security architectures, not just with RBAC.

However the solution poses few limitations, which needs to be worked further on. Among them the most significant are:

- Hard to determine exact context - sometimes can happen that some resource should be accessible just from given context. For example some resources are accessible only during the day and some just during the night. Such scenario is impossible to secure with proposed solution.
- Levels are linear - structure of the levels is strictly linear and therefore it is impossible to build some tree or even more complex structure of levels. Often happen that there are multiple context rules, which are granted different set of right. Levels can't model for example geographical situation when users from same state have some rights but people in different location of the state got additional specialized rights.

5. Conclusion

1 - Jak se to ma ted

2 - Co jsem vymyslel

3 - Co chci zlepsit dal

Acknowledgements

Research described in the paper was supervised by Prof. A. Supervisor, FEE CTU in Prague and supported by the Czech Grant Agency under grant No. 102/ 01/9999, by the Czech Ministry of Education under grant No. 9999/2002 and by the research program MSM 444222.

The headline *Acknowledgements* is of the style `\section*{}`.

The authors are asked to pay special attention to the form of references. The NAMES OF AUTHORS should be typed in capitals, the *Titles of Journals, Books or Proceedings* in italics with the first capital letter in all significant words. The tittles of articles are typed similarly as the basic text without the first capital letter in all words. Use the standart environment `thebibliography`.

References

[1] Matthew J Moyer, and M Abamad. Generalized role-based access control. In: Dis- tributed Computing Systems, 2001. 21st International Conference on.. 2001. 391 398.

[2] Michael J Covington, Wende Long, Srividhya Srinivasan, Anind K Dev, Mustaque Ahamad, and Gregory D Abowd. Securing context-aware applications using en- vironment roles. In: Proceedings of the sixth ACM symposium on Access control models and technologies. 2001. 1020.

[3] Seon-Ho Park, Young-Ju Han, and Tai-Myoung Chung. Context-role based access control for context-aware application. 2006.

[4] Goran Sladi/'c, Branko Milosavljevi/'c, and Zora Konjovi/'c. Context-sensitive access control model for business processes. Computer Science and Information Sys- tems/ComSIS. 2013, 10 (3), 939972.

[5] Devdatta Kulkarni, and Anand Tripathi. Context-aware role-based access control in pervasive computing systems. In: Proceedings of the 13th ACM symposium on Access control models and technologies. 2008. 113122.

[6] Gustaf Neumann, and Mark Strembeck. An approach to engineer and enforce con- text constraints in an RBAC environment. In: Proceedings of the eighth ACM sym- posium on Access control models and technologies. 2003. 6579.

[7] Ghita Kouadri Mostéfaoui, and Patrick Brézillon. A generic framework for context- based distributed authorizations. 2003.

[8] A Corrad, Rebecca Montanari, and Daniela Tibaldi. Context-based access control management in ubiquitous environments. In: Network Computing and Applications, 2004.(NCA 2004). Proceedings. Third IEEE International Symposium on. 2004. 253260.

[9] Joao Carlos D Lima, Cristiano C Rocha, Iara Augustin, and Mrio AR Dan- tas. A Context-Aware Recommendation System to Behavioral Based Authentication in Mobile and Pervasive Environments. In: Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on. 2011. 312319.

[10] Le Xuan Hung, J Hassan, AS Riaz, SMK Raazi, Y Weiwei, Ngo Trong Canh, Phan Tran Ho Truc, Sungyoung Lee, Heejo Lee, Yuseung Son, and others. Activity- based security scheme for ubiquitous environments. In: Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International. 2008. 475481.

[11] Zhang Wendong, and Zhang Kaiji. A role-based workflow access control model. In: Education Technology and Computer Science, 2009. ETCS09. First Interna- tional Workshop on. 2009. 11361139.

About Authors...

First AUTHOR was born in ... The biography is typed using the environment `authorcv{First AUTHOR}`.

For one author, one paragraph of the biography is devoted. The **Name** of the author is typed in `\textbf{}`, the **SURNAME** is written in capitals. All authors must have a student status!