Poster: Reactive Access Control Systems

Maryam Davari
Purdue University
West Lafayette, Indiana
davari@purdue.edu

Elisa Bertino
Purdue University
West Lafayette, Indiana
bertino@purdue.edu

ABSTRACT

In context-aware applications, user's access privileges rely on both user's identity and context. Access control rules are usually statically defined while contexts and the system state can change dynamically. Changes in contexts can result in service disruptions. To address this issue, this poster proposes a reactive access control system that associates contingency plans with access control rules. Risk scores are also associated with actions part of the contingency plans. Such risks are estimated by using fuzzy inference. Our approach is cast into the XACML reference architecture.

KEYWORDS

Context-aware applications, reactive access control system, contingency plan, risk, Fuzzy inference, XACML

ACM Reference Format:
Maryam Davari and Elisa Bertino. 2018. Poster: Reactive Access Control Systems. In *SACMAT '18: The 23rd ACM Symposium on Access Control Models & Technologies (SACMAT), June 13–15, 2018, Indianapolis, IN, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3205977.3208947

1 INTRODUCTION

Recent access control models and standards, such as XACML, are context based. A context-based access control system uses context (i.e., a condition or requirement on the current state and attributes of subjects, protected objects, and their environments) to evaluate access requests. Well known context-based access control models include T-RBAC [6] and GEO-RBAC [13] that consider, respectively, time and location as the contextual information relevant for access control. In a context-based access control system, rules determine which access prohibitions or permissions can be applied with respect to specific circumstances. For example, in a health care domain, physicians have different permissions in particular contexts (e.g., "urgency", "industrial medicine") [11].

In a pervasive environment, users are mobile and access resources (e.g., services, sensors) that utilize mobile devices; in addition, such environments continuously evolve. A critical issue is that the context of the subject (e.g., location, time, network state, network security configuration) can dynamically change. Thus accesses that were permitted based on certain contextual conditions may not any longer be permitted. If such a change were not anticipated, the accesses may have to be blocked and this may result in severe application disruptions. To address such problem, mechanisms are required to minimize service disruption.

To address such need, we propose the notion of a reactive access control that extends the context-based access control mechanism in order to handle unexpected context changes. We define an event as an activity that occurs in time (e.g., leaving a room) and can change the behavior of the system. Anticipating all events is not easy. Therefore, we use the contingency plan concept to handle the events. A contingency plan finds an alternative decision to deal with unexpected events and it is separated from a normal operation. However, a contingency plan can have a potential to add risks to the system. Hence, an appropriate action should be selected such that to keep the risks under control.

We cast our model into XACML [1] by adding to it a specialized component referred to as reactive-XACML (R-XACML, for short) - see Figure 1.

2 OVERVIEW OF THE PROPOSED APPROACH

Assume a scenario in which an access request has been already permitted; then the context changes and the request cannot be completed as the rules under which the access had been permitted are not applicable in the new context. Suppose also that the PDP cannot find any rule that allows the access to continue. In such a case, instead of sending an immediate service interruption notification to the application, the PDP sends notifications to the PIP and the Contingency Plan Manager (CPM). The PIP computes contingency plan parameters (described in the following subsection) and pushes them to the CPM. We use the push-based model (used in the reactive programming) [2] to automatically propagate new attribute values to the CPM when an event occurs.

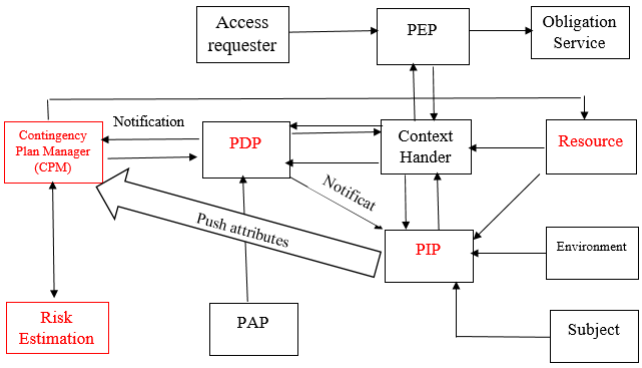

Figure 1: Architecture of R-XACML

SACMAT '18, June 13–15, 2018, Indianapolis, IN, USA

ACM ISBN 978-1-4503-5666-4/18/06.
https://doi.org/10.1145/3205977.3208947

The current access continues temporarily until receiving a response from the CPM. The CPM either allows the current access to be completed or proposes an alternative action. Then, the action is submitted to the risk estimation component to estimate the risks associated with the action. If the estimated risk is low, the action will be submitted to the PDP. Otherwise, the interaction between the CPM and risk estimation is repeated until identifying an appropriate action with a low risk. The CPM and risk estimation components are described in the following subsection in more details.

2.1 Contingency Plan Manager (CPM)

Policies are specified under assumptions about environments and events. The applications can do certain actions when specific context conditions hold. The question is "what if" the determined actions cannot complete in time. The importance of developing alternative actions (invoked by contingency plans) in access control application is undeniable as shown by the following example.

In an American health care center, medical records can be read only during office hours and from inside of the USA but what if an access is requested from another country and outside of office hours. Assume that a patient has some medical records managed by this health care center and that he travels to Spain for a couple of days. Suppose that he is badly hurt in a car crash and taken to a hospital. An emergency doctor needs to view the medical records of the patient to check if he has allergies to specific substances. The access request is submitted from outside of the USA and office hours. According to the mentioned health care policy, the doctor is not allowed to access the patient's medical record. A contingency plan is required (e.g., giving the reading access to the emergency doctor after validating the identity of the emergency doctor and notifying the patient's physician).

Due to the importance of alternative actions especially for access control applications, we develop the CPM component for XACML. The CPM contains three main components supporting event specification, event detection, and contingency plans for events (P1-P3) described as follows:

P1. Event specification: Events can be classified as the primitive events (a single event) and composite events (the disjunction, sequence, or conjunction of a set of primitive events) based on the number of events happing synchronously [16].

P2. Event detection: The predefined events are recognized from the contexts and stored in a chronological order.

P3. Contingency plans for events: After identifying involved events in the request, the CPM follows five steps to make a final decision.
Step 1. Parameter set: The CPM defines a set of parameters including the type of event (primitive, composite), type of each involved primitive event (temporal, non-temporal), type of action, type of resource, time of event occurrence, action start time, and action completion time. The parameter values are automatically collected by the PIP from the current request by using the push-based model. Then, the parameter values are sent to step 2 for the evaluation.
Step 2. Event evaluation: The CPM checks if the event is temporary by waiting (using the system clock) and monitoring the event. If the event value becomes acceptable (by the rules of the application), it resumes continuing the current action (i.e., the action before interruption). Also, for rules with composite events, decisions are made base on the number of events with acceptable values and the value of the most effective event. If it can satisfy the rule, the access resumes. Otherwise, it goes to step 3.
Step 3. Event occurrence time comparison: The time of event occurrence is compared with the action start time and estimated action completion time. If the time of event occurrence is almost closed to the estimated completion time, the access continues. While if the event occurs right after the start of the action, step 4 is followed.
Step 4. Resource sensitivity identification: Some resources are more sensitive than others (e.g., sensitive medical information). If a resource is not sensitive, the access is granted. However, for sensitive information, the type of action is required to make a final decision. Step 5 is followed for this purpose.
Step 5. Action Type recognition: In access control systems, most of the actions are database operations including "read", "insert", "delete", "update", etc. They take a finite amount of time for execution. If the access does not change the data, the event is ignored and the system resumes the action. However, for actions changing the data (including insert, delete, and update), the CPM gives only the access to the copy of the resource. In this case, when a new request is received from the user, the user's confirmation of changes is required. In the case of discarding the changes by the user, the actual data is not updated.

2.2 Risk Estimation

Contingency plans can be associated with risks. Assume a government building containing sensitive information is on fire while firemen are waiting for the security clearance (that is not received in time). In this case, a contingency plan is essential that can let the firemen access the building after signing a non-disclosure agreement. However, the risk resulting from the given access can be high. Contingency plans should be chosen wisely to minimize risks.

The risk estimation predicts the probability of risks incurred by employing contingency plans. One appropriate solution for the risk assessment implementation is fuzzy inference. This technique has been employed in many areas, including medicine, engineering, management application [17]. To implement this technique, first, we collect previous experience containing a set of rule and risk factor information (e.g., an experience shared by system administrators and security researchers). Second, the vague concepts (e.g., low, high) in the collected experience are explained by membership functions in fuzzy inference. Third, fuzzy rules can be any arbitrary function. Actual risk estimation functions can be used in the case of existence.

3 RELATED WORK

3.1 Context-Aware Access Control

Several context-based access control models have been proposed. DRBAC [10] adjusts roles and permission assignments dynamically according to the context information. CA-RBAC [19] includes temporal and spatial constraints by using context-based constraints. GTRBAC [6] uses temporal constraints to enable and disable roles.

Or-BAC [14] introduces abstractions for actions and objects named activities and views, respectively. In Or-BAC, contextual information can be specified to manage given contexts. Despite the importance of alternative actions especially for all time-constraint applications, none of the mentioned approaches consider the use of contingency plans.

An emergency access control model that can manage information sharing was proposed by Carminati et al. [5]. It uses Complex Event Processing (CEP) systems (e.g., Oracle CEP) to automatically detect emergency situations from the contextual information. Break-glass [9], [3] is another approach to handle emergency situations. The Break-glass access control model allows subjects to override decisions without considering the reason for request denial. Rumpole [15] enhances the traditional Break-glass model by checking if the user is allowed to override denial decisions according to accepted obligations. These paradigms only focus on emergency situations. However, the reactive access control system can handle unexpected context changes in different situation.

3.2 XACML

Approaches have been proposed for testing, analyzing, and modeling XACML policies [18], [12]. Such approaches introduce environmental and contextual roles (i.e., roles which are activated/deactivated as a result of an event occurrence in the system). Several formal models, such as the ones based on description logics [4] and model-oriented specification languages [8], and verification techniques [7] have been proposed for XACML. To the best of our knowledge, however, no approach has been proposed for dealing with context changes in XACML.

4 CONCLUSION

In context-aware applications, access requests are permitted based on the contexts of requests. If the context of an access request cannot satisfy the access requirements in the time-constraints, the access will be blocked. To address this issue, we propose the notation of reactive access control mechanism by extending the traditional context-aware applications. We present the reactive XACML in which the contingency plans handle the context changes in the system. In the R-XACML, the user's access is continuously adapted for the contingency environments. The risks associated with the contingency plans are also calculated by using fuzzy inference to keep risks as low as possible.

ACKNOWLEDGMENTS

The work reported in this paper has been partially supported by the U.S. Army Research Laboratory and the U.K. Ministry of Defense under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defense or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] Anne Anderson, Anthony Nadalin, B Parducci, D Engovatov, H Lockhart, M Kudo, P Humenn, S Godik, S Anderson, S Crocker, et al. 2003. extensible access control markup language (xacml) version 1.0. *OASIS* (2003).
[2] Engineer Bainomugisha, Andoni Lombide Carreton, Tom van Cutsem, Stijn Mostinckx, and Wolfgang de Meuter. 2013. A survey on reactive programming. *ACM Computing Surveys (CSUR)* 45, 4 (2013), 52.
[3] Achim D Brucker and Helmut Petritsch. 2009. Extending access control models with break-glass. In *Proceedings of the 14th ACM symposium on Access control models and technologies*. ACM, 197–206.
[4] Jeremy W Bryans and John S Fitzgerald. 2007. Formal engineering of XACML access control policies in VDM++. In *International Conference on Formal Engineering Methods*. Springer, 37–56.
[5] Barbara Carminati, Elena Ferrari, and Michele Guglielmi. 2011. Secure information sharing on support of emergency management. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. IEEE, 988–995.
[6] Yehia ElRakaiby, Frederic Cuppens, and Nora Cuppens-Boulahia. 2008. Interactivity for reactive access control. *Sar Ssi 2008* (2008), 257.
[7] Kathi Fisler, Shriram Krishnamurthi, Leo A Meyerovich, and Michael Carl Tschantz. 2005. Verification and change-impact analysis of access-control policies. In *Proceedings of the 27th international conference on Software engineering*. ACM, 196–205.
[8] Daniel Jackson, Ilya Shlyakhter, and Manu Sridharan. 2001. A micromodularity mechanism. In *ACM SIGSOFT Software Engineering Notes*, Vol. 26. ACM, 62–73.
[9] NEMA Joint. 2004. COCIR/JIRA Security And Privacy Committee (SPC). *Break-glass: An approach to granting emergency access to healthcare systems* (2004).
[10] James BD Joshi, Elisa Bertino, Usman Latif, and Arif Ghafoor. 2005. A generalized temporal role-based access control model. *IEEE Transactions on Knowledge and Data Engineering* 17, 1 (2005), 4–23.
[11] Anas Abou El Kalam, R El Baida, Philippe Balbiani, Salem Benferhat, Frédéric Cuppens, Yves Deswarte, Alexandre Miege, Claire Saurel, and Gilles Trouessin. 2003. Organization based access control. In *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*. IEEE, 120–131.
[12] Vladimir Kolovski, James Hendler, and Bijan Parsia. 2007. Analyzing web access control policies. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 677–686.
[13] Devdatta Kulkarni and Anand Tripathi. 2008. Context-aware role-based access control in pervasive computing systems. In *Proceedings of the 13th ACM symposium on Access control models and technologies*. ACM, 113–122.
[14] Kim Tuyen Le Thi, Tran Khanh Dang, Pierre Kuonen, and Houda Chabbi Drissi. 2012. STRoBAC–spatial temporal role based access control. In *International Conference on Computational Collective Intelligence*. Springer, 201–211.
[15] Srdjan Marinovic, Robert Craven, Jiefei Ma, and Naranker Dulay. 2011. Rumpole: a flexible break-glass access control model. In *Proceedings of the 16th ACM symposium on Access control models and technologies*. ACM, 73–82.
[16] Deepak Mishra. 1991. *SNOOP: an event specification language for active database systems*. Master's thesis. University of Florida.
[17] Qun Ni, Elisa Bertino, and Jorge Lobo. 2010. Risk-based access control systems built on fuzzy inferences. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. ACM, 250–260.
[18] Que Nguyet Tran Thi and Tran Khanh Dang. 2012. X-STROWL: A generalized extension of XACML for context-aware spatio-temporal RBAC model with OWL. In *Digital Information Management (ICDIM), 2012 Seventh International Conference on*. IEEE, 253–258.
[19] Guangsen Zhang and Manish Parashar. 2004. Context-aware dynamic access control for pervasive applications. In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*. 21–30.