Dynamic Access Control Framework for Enterprise Content Management Systems

Nadia Hocine and Ismail Bokhari
University of Abdelhamid Ibn Badis of Mostaganem, Mostaganem, Algeria

Keywords: Access Control, Enterprise Content Management System.

Abstract: With the large adoption of telework business model, employees can work anytime, anywhere and sometimes with their own personal devices due to the limited financial capabilities of their companies. Many issues of the security and access control of remote exchanges of Enterprise Content Management systems (ECM) have to be considered. In particular, the access control should be adapted to employees' context, their multiple device capabilities as well as their profiles and situations to increase the usability of the system. However, most access control models do not take into account the users' profiles and the variability of their devices in open network. They also focus on the continuous intervention of administrators to manage the system and add new devices and set parameters. With the diversity of users' devices and context conditions in telework, access control needs to be dynamically managed to reduce human intervention. In this paper, we suggest an agent-based access control framework that focuses on M5StickC external device used as an access badge. The framework is based on a multi-level rule engine to dynamically generate policies according to users' context, profile and device. It is implemented and proposed as an open-source solution for small companies to manage their own ECM access control.

1 INTRODUCTION

Telework becomes a popular alternative to work that consists in relocating work to reduce costs while increasing the well-being and the safety of employees. It may also be an alternative solution for employees with different profiles and high mobility. Managing remotely enterprise resources and their access can be challenging when employees work with their personal devices, under mobility condition (such as home, airport and train) and with particular disabilities. Therefore, the access control management should take into account not only the context and user various devices but also their profile in terms of work and health conditions in order to enhance the usability of the access control applications.

Diffrent dynamic access control strategies have been proposed in the literature to deal with dynamic nature of user environments and contexts (Freudenthal et al., 2002) (Calo et al., 2018) (Oluwatimi et al., 2018). However, most access control models do not take into account the variability of users devices in open network as well as usability issues. Moreover, most strategies require human administrator intervention (Calo et al., 2018) which can be a challenging task, especially in the case of variability of devices and users' profile.

In this paper, we suggest a dynamic access control framework that focuses on an agent-based system. Providing users with an intelligent system can improve not only the management of contextual data for access control in a lightweight decentralized infrastructure (Uddin, 2019), but also allows more personalized access control that deals with usability issues. Indeed, with their capability to think and make decision, software agents can control the dynamic context information and can help in making decision on resources access authorization. Agents use a rule-based system to set resources permissions. In fact, an access control system is based on the expert decisions on knowledge constituted from various data on users and their context information that can be represented using rules (Bădică et al., 2011). However, a rule-based system may require high level computing performance with a large number of rules, known as the redundancy issue (Bădică et al., 2011). To deal with this issue, our proposed framework considers a multi-level rule engine that selects only a set of rules to generate dynamically an access control policy on the basis of resources' sensibility.

The rest of this paper is organized as follows: Section 2 discusses related works on dynamic access con-

Hocine, N. and Bokhari, I.
Dynamic Access Control Framework for Enterprise Content Management Systems.
DOI: 10.5220/0010514806470652
In *Proceedings of the 18th International Conference on Security and Cryptography (SECRYPT 2021)*, pages 647-652
ISBN: 978-989-758-524-1

trol techniques. In section 3, we introduce our proposed agent-based control framework. We describe in this section the architecture of the proposed framework as well as the policy specification and generation. Section 4 presents the implementation details and the test of the proposed framework. Finally, Section 5 concludes the paper and outlines our future works.

2 RELATED WORKS

Access control is the verification of whether user actions on resources are authorized according to a security policy (Sandhu and Samarati, 1994). It allows controlling the users' access to sensitive resources (Nyakomitta and Abeka, 2020). Different access control approaches have been proposed in the literature according to specific systems.

Role-based access control approach has been initially adopted in various institutions and companies due to its simplicity in large-scale authorization management (Ferraiolo et al., 2016). This approach uses a model that considers three concepts: users, roles and permissions. Roles are users' job functions within the organizations and permissions are the approvals to perform certain operations on resources. The access permissions are assigned to roles in order to manage easily permissions (Liu et al., 2017). However, the role based policy model does not consider context constraints such as user location and devices.

Dynamic access control approaches can be used to deal with the flexible control of resources access permissions according to available context information (Corradi et al., 2004). For instance, different works extended the role-based approach by considering temporal and spatial information (Joshi et al., 2005) (Damiani et al., 2007) as well as the resource and environment dimensions (Hosseinzadeh et al., 2016) (Trnka and Cerny, 2016). They focused on the dynamic user-role and role-permission assignments through contextual constraints, called dynamic attributes (Zheng et al., 2011). For example, Kulkarni and Tripathi proposed a programming framework that extended the role-based access control model by considering user and resource conditions in role-permission assignments (Kulkarni and Tripathi, 2008). In Trnka and Cerny' work, user roles and context information such as the IP addresses and times of the day were used to determine resources authorization (Trnka and Cerny, 2016). In addition to temporal and spatial constraints, access control can also consider different attributes and real-time user's situations. For instance, Kayes et al. proposed a situation-aware access control framework based on an ontology (Kayes et al., 2019). A policy model was defined using conditional expression of users' situations. In the previous reviewed works, the user-role assignment policies related to dynamic context constraints require the continuous administrator intervention. They also depend on context data regardless users' profiles and their devices capabilities and variability.

With the technological advancement of devices and their embedded sensors, various approaches were based on users' devices to control their own access to resources (Squicciarini et al., 2009) (Calo et al., 2018). For instance, Verma et al. suggested a framework in which the manager uses an interaction graph to define the roles that devices may play in the system (Verma et al., 2017). Other works considered the networked environment model in order to give the possibility of each device to obtain security constraints of other devices of its environment (Squicciarini et al., 2009). However, this approach requires the deployment of an environment model specification that should be supervised by administrators which can be, in turn, a challenging task.

The previous reviewed access control models are often based on centralized management of permissions by an authorization entity which can reduce its performance with the increase of number of users and requests. Existing dynamic access control models can difficult to execute in distributed systems that have a large number of users and/or ressources. The centralized management of access control in distributed systems may cause access delay or unauthorized access because of the server overload. Moreover, due to its centralized nature, it can become easily the target of network malicious attacks.

To deal with the previous issues, a distributed management of access control can be used. It consists in allowing each node of the network to control the dynamic mapping between users and permissions to improve the access control management performance. It can be also used to personalize the access control to users with various devices, profiles and situations without requiring continuous administrator intervention. In this paper, we propose an agent-based access control framework for Entreprise Content Management systems (ECM). Agent-based systems can play an important role in building distributed intelligent access control management frameworks that take into account various user device, context and profiles. Unlike dynamic access control models that not consider open network due to vulnerability issues and excessive intervention of administrators, we suggest a distributed access control management that focuses

on an agent-based architecture. We also provide users with an external "access badge" to initiate and obtain their access permissions. The objective is to deal with the variability of user devices and their limited capabilities in terms of energy and sensors disposition. In order to limit vulnerabilities in the case of mobility, the access request is limited to Bluetooth connexion with small amplitude parameters and within limited time. Next, we describe the proposed framework as well as the dynamic policy generation for ECM.

3 AN AGENT-BASED ACCESS CONTROL FRAMEWORK FOR ECM SYSTEMS

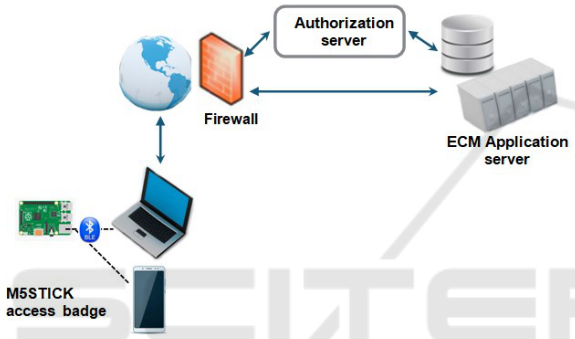

Figure 1: General framework.

In this paper, we suggest an access control framework for enterprise content management systems (ECMs). ECM is a software program that consists of processes, procedures, and technologies used in conjunction to manage unstructured content such as documents and files of enterprise (Katuu, 2018). It helps employees upload resources that are identified by their meta-data attributes. The latter are used in access control and can be classified according to: resource owners, resource type and category as well as resource sensitivity.

With the rapid evolution of telework adoption in our society nowadays, employees may use various personal devices to communicate and get services of the ECM anywhere and anytime. Different contextual information, such as user location, request time and mobility can be used to enforce access control and system security. Contextual information are controlled through independent services and programs that act as intelligent agents and help the development of more intelligent access control systems (Uddin, 2019) (Calo et al., 2018).

The proposed agent-based access control framework is dedicated to small companies that need the personalization of access control by taking into account users' context and conditions while limiting administrator intervention. As shown in Figure 1, the user is provided with a M5STICK device used to obtain access authorization from user device (such as laptop and phone). M5STICK device includes various features and sensors such as Bluetooth 4.0, Wifi, Accelerometer, etc. Using his or her M5STICK badge, an employee can acquire permissions with a Wifi or Bluetooth connection to his or her telework device. We used Bluetooth connection in this work. Indeed, a Bluetooth-based infrastructure makes the system more resilient to malicious attacks especially in user mobility conditions (Wong and Hunter, 2017).

We used Arduino development environment in order to manage Bluetooth connection with devices and to obtain contextual data. After receiving the access request, the authorization server will be responsible on generating the policy using a rule-based system. Then, once the permission is accorded, the user can obtain access to resources using the ECM application server. We used in this paper Alfresco bitnami VM (Pal, 2016) as an example of ECM system. Next, we describe the architecture of the framework as well as the policy specification and generation.

3.1 Access Control Framework Architecture

The framework focuses on an agent-based architecture (see Figure 2). Agents are software entities that can send and receive data from sensors or from other agents in a particular environment. According to Russell "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors" (Russell and Norvig, 2002). In the proposed framework, agents are used to represent software entities that control the user contextual data as well as to ensure the internal functioning of the access control through policy generation. It consists of: the user agent, authorization server that includes an agent policy generator as well as an authoring tool.

- User Agent. It is responsible on communicating with the authorization server to acquire the access policy. It also provides the latter with context information from sensors. In fact, to obtain permissions on his or her laptop or smartphone, the user has to obtain access authorization through a M5STICK badge. Once the access is permitted, the user can communicate with the ECM application server to get authorized resources using his own User Agent.

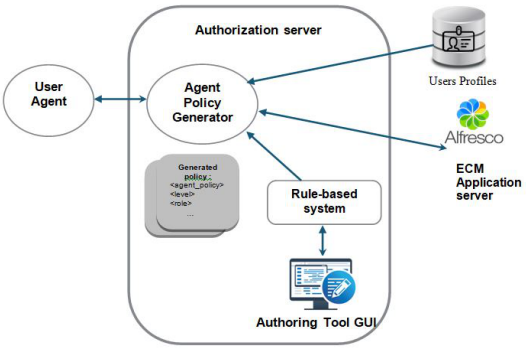

Figure 2: Framework architecture.

- Authorization Server. It aims to control dynamically the user permissions on resources by generating appropriate policy and providing it to the user. It consists of the following elements: rule based system to represent the expert decisions on knowledge constituted from various data on users and their context information, the authoring tool used by the administrator to introduce general permission constraints and the agent policy generator that receives users' access request though their User Agents and generates policies.
- The Application Server. It allows performing actions on resources of the enterprise content management system. The access permissions accorded to the user follow the policy generated by the authorization server.
- User Profile. Each user of the system is characterized by a profile which represents a set of attributes related to the user such as: age, mobility, disabilities, kind of contract, etc. It is used to personalize access control constraints to user specific conditions.

3.2 Policy Generation

Policy specification may depend on different languages. For example, the role-based access control policy may be expressed using access control lists (ACL) to resources stored in a central or distributed memory (Verma et al., 2017). It is a simple way to map resources to permissions while considering resources hierarchy. However, one of the limitations of ACLs is the low expressiveness of regular expressions for dynamic contextual data and meta-data based resources identification.

XML (extensible markup language) based policy specification has been largely used due to its extensible nature and simplicity to specify policy expression with different data types. Extensible access control markup language or XACML was proposed initially for role-based approach and has then been extended and used in various works (Ferraiolo et al., 2016). Using XACML to encode the policy consists in the definition of expressions that return values that reflect the kind of authorization or deny of access.

In this paper, we focus on XML-based policy language inspired by the XACML specifications. Unlike existing policy generation techniques, we focus on a rule-based generation that depends on each device and user attributes. It also can be initiated by the administrator by introducing only general rules using an authoring tool that does not require specific technical skills. The policy XML file generation is based on expert knowledge through a rule based system to infer the decision on resources permission. Using Markup Scheme such as XML also allows the communication between agents and data transfer via a network.

```
<Agent_policy>
    <Role roleID= "id"  name="name">
        <Metadata classification="classification" type="application_type"/>
        <Levels>
            <Level id=" level_id" model=" ProfileBasedAccessControl">
                <Constraints>
                    <Profile_constraint id="id" >
                        <arg> value </arg>
                        <permission  value="allow"/>
                    </Profile_constraint>
                </Constraints>
            </Level>
            <Level id="level_id" model= "ConstraintBasedAccessControl">
                <Constraints>
                    <Context_constraint id=" id">
                        <arg> value </arg>
                        <permission value="allow"/>
                    </Context_constraint >
                </Constraints>
            </Level>
        </levels>
    </Role>
</Agent_policy>
```

Figure 3: Policy specification.

The Agent Policy Generator uses the rule-based system to generate a policy by mapping user-agents to permissions. However, a rule-based system may require high level computing performance due to the large number of rules, known as the redundancy issue. We propose therefore a personalized access control that reduces the number of rules used to generate dynamically a policy. Indeed, the Agent Policy Generator selects only a small set of eligible rules according to a security level that depends on resources sensibility. For instance, according to the resources low level of sensibility, the agent can exclude for instance rules that consider the context information, such as working time and location.

According to resources sensibility, three rules levels were considered in our framework. The first level, consists of role-based rules that represent the constraints that use agent-role mapping without considering context and profile information. The second level is based on user profile data such as availability and health conditions. The third level considers the contextual information through the dynamic environment characteristics mapping with resources classification.

The Policy Generator Agent generates an XML file with the appropriate policy that uses the general specification shown in Figure 3.

4 IMPLEMENTATION

The project repository can be obtained from Github, with this link: https://github.com/AuthorisationTool We have performed an initial test of the framework using M5StickC device and two laptops. The devices used were Intel I7-7500U, 8 GB DDR4 RAM, SSD 240 GB. M5StickC device is an internet of things development card based on ESP32 micro-controller used in our framework as an access badge. It consists of various modules especially: Bluetooth 4.0, Wifi, Gyro-meter, Accelerometer, Microphone, USB, IR Transmitter, LCD 0,96" screen and 3 Buttons.

Each badge characteristics are stored in a PostgreSQL database of the authorization server. This includes the identifier of the badge and the identifier of the process HID (Holder ID) used to identify the user. Using a Bluetooth 4.0 connection, the user can obtain access from his or her telework device. The server of the badge was based on GATT protocol and Google Chrome as navigator. Figure 4 shows the badge used in our first test.

Figure 4: M5StickC device.

```
<Agent_policy>
    <Role roleID= "1"  name="role1">
        <Metadata classification="type" type="web-application"/>
        <Levels>
            <Level id="1" model=" ProfileBasedAccessControl">
                <Constraints>
                    <Health id="HE1" >
                        <arg> Healthy </arg>
                        <permission  value="allow"/>
                    </Health>
                </Constraints>
            </Level>
            <Level id="2" model= "ConstraintBasedAccessControl">
                <Constraints>
                    <Agent_location id=" AL1">
                        <arg> Home </arg>
                        <permission value="allow"/>
                    </Agent_location>
                    <Acceleration id=" AC1">
                        <arg> 1 3 1 3 1 3 </arg>
                        <permission value="allow"/>
                    </Acceleration>
                </Constraints>
            </Level>
        </levels>
    </Role>
```

Figure 5: Example of generated policy.

To generate policies, rules were implemented through a simple rule engine that we developed. An example of generated policy is shown in Figure 5. The profile constraint that we have considered in our first test was: "Health" which is a list of abstract health conditions of the user such as "Healthy" and "Disability". The latter determines personalized time and spatial conditions to be considered for specific user's health conditions. As for the context data, three constraints have been included in the test : (i) "User Agent location" which represents the geographical site position that includes values on latitude, longitude and diameter of user location (ii) "Gyro" which considers user movement data that includes the speed recorded when the user asks access permission (iii) "Acceleration" of the agent movement.

We developed an authoring tool that help administrators easily set and update the general rules of access control without requiring high development skills. Using the authoring tool Web interface, the administrator set first the possible ressources types, category, sensitivity levels and owners for role creation. The ressources configuration follows the possible considerations of Alfresco ECM. Once the role is created, the administrator can add levels that include context and/or profile constraints. The administrator can also add his or her personalized constraints or uses the default possible constraints.

The results of the first test with two experts showed the simplicity of the interface and efficiency of the access badge to get rapid and easy access to the Alfresco server application.

5 CONCLUSION

In this paper, we presented an agent-based access control framework architecture for enterprise content management systems. An open source tool is proposed following this architecture to help small companies in controlling access to ECM ressources while limiting administrator's intervention. The tool is based on a rule engine that generates dynamically an access control policy on the basis of users' conditions and context information obtained with the help of an M5StickC access badge. Furthermore, we proposed an authoring tool that allows administrators to introduce general constraints without having strong knowledge about the rule and policy specification. The results of the initial test of the tool with two experts showed the simplicity of the interface and efficiency of the access badge to get rapid and easy access to the Alfresco server application. In our future works, we plan to test the performance of this tool

with a large number of employees and administrators through different case studies. The objective is to evaluate the efficiency of the tool to improve both authoring tool and policy generation. Finally, we plan to test the usability of the system with employees with particular health conditions (such as post-stroke patients) by considering various profile-based rules that include personalized spatial and temporal constraints.

REFERENCES

Bădică, C., Braubach, L., and Paschke, A. (2011). Rule-based distributed and agent systems. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 3–28. Springer.

Calo, S., Verma, D., Chakraborty, S., Bertino, E., Lupu, E., and Cirincione, G. (2018). Self-generation of access control policies. In *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*, pages 39–47.

Corradi, A., Montanari, R., and Tibaldi, D. (2004). Context-based access control for ubiquitous service provisioning. In *Proceedings of the 28th International Computer Software and Applications Conference*, pages 444–451. IEEE.

Damiani, M. L., Bertino, E., Catania, B., and Perlasca, P. (2007). Geo-rbac: a spatially aware rbac. *ACM Transactions on Information and System Security*, 10(1).

Ferraiolo, D., Chandramouli, R., Kuhn, R., and Hu, V. (2016). Extensible access control markup language (xacml) and next generation access control (ngac). In *International Workshop on Attribute Based Access Control*, pages 13–24.

Freudenthal, E., Pesin, T., Port, L., Keenan, E., and Karamcheti, V. (2002). drbac: distributed role-based access control for dynamic coalition environments. In *Proceedings 22nd International Conference on Distributed Computing Systems*, pages 411–420. IEEE.

Hosseinzadeh, S., Virtanen, S., Díaz-Rodríguez, N., and Lilius, J. (2016). A semantic security framework and context-aware role-based access control ontology for smart spaces. In *Workshop on Semantic Big Data*, pages 1–6.

Joshi, J. B., Bertino, E., Latif, U., and Ghafoor, A. (2005). A generalized temporal role-based access control model. *IEEE transactions on knowledge and data engineering*, 17(1):4–23.

Katuu, S. (2018). A comparative assessment of enterprise content management maturity models. In *E-manufacturing and e-service strategies in contemporary organizations*, pages 93–118. IGI Global.

Kayes, A., Han, J., Rahayu, W., Dillon, T., Islam, M. S., and Colman, A. (2019). A policy model and framework for context-aware access control to information resources. *The Computer Journal*, 62(5):670–705.

Kulkarni, D. and Tripathi, A. (2008). Context-aware role-based access control in pervasive computing systems. In *Access control models and technologies*, pages 113–122.

Liu, Q., Zhang, H., Wan, J., and Chen, X. (2017). An access control model for resource sharing based on the role-based access control intended for multi-domain manufacturing internet of things. *IEEE Access*, 5:7001–7011.

Nyakomitta, P. S. and Abeka, S. O. (2020). Security investigation on remote access methods of virtual private network. *Global Journal of Computer Science and Technology*.

Oluwatimi, O., Damiani, M. L., and Bertino, E. (2018). A context-aware system to secure enterprise content: Incorporating reliability specifiers. *Computers & Security*, 77:162–178.

Pal, V. (2016). *Alfresco for Administrators*. Packt Publishing Ltd.

Russell, S. and Norvig, P. (2002). Artificial intelligence: a modern approach.

Sandhu, R. S. and Samarati, P. (1994). Access control: principle and practice. *IEEE communications magazine*, 32(9):40–48.

Squicciarini, A. C., Shehab, M., and Paci, F. (2009). Collective privacy management in social networks. In *Proceedings of the 18th international conference on World wide web*, pages 521–530.

Trnka, M. and Cerny, T. (2016). On security level usage in context-aware role-based access control. In *Proceedings of the 31st ACM Symposium on Applied Computing*, pages 1192–1195.

Uddin, I. (2019). *A rule-based framework for developing context-aware systems for smart spaces*. PhD thesis, University of Nottingham.

Verma, D., Calo, S., Chakraborty, S., Bertino, E., Williams, C., Tucker, J., and Rivera, B. (2017). Generative policy model for autonomic management. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing*, pages 1–6. IEEE.

Wong, K. and Hunter, A. (2017). Bluetooth for decoy systems: A practical study. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 86–387. IEEE.

Zheng, J., Zhang, Q., Zheng, S., and Tan, Y. (2011). Dynamic role-based access control model. *Journal of software*, 6(6).