TOWARDS A SCALABLE AND DYNAMIC ACCESS CONTROL SYSTEM FOR WEB SERVICES

Meriam Jemel, Nadia Ben Azzouna and Khaled Ghedira
Research laboratory SOIE (Stratégies d'Optimisation et Informatique intelligentE)
41 Rue de la Liberté, Cité Bouchoucha, 2000 Le Bardo, Tunis, Tunisia

Keywords: Web Service's Access Control, Context-awareness, Scalability, Multi-agent Systems, Simulation.

Abstract: Web services are vulnerable to different types of security attacks. The problem of secure access to web-based applications is becoming increasingly complex. Management complexity arises because of the scalability considerations such as the large number of web services users and their invocations and the fact that the access control system should take into account the context. In this paper we describe the architecture of our TDRBAC (Trust and Dynamic Role Based Access Control) model which is implemented using agent technology. In fact, this technology fulfills several requirements of web service's access control by providing both context awareness and scalability. In order to verify the scalability of the proposed solution, we expose some experimental results from a prototype implemented using JADE (Java Agent DEvelopment) platform. The performance tests show that our TDRBAC multi-agent based system meets the scaling requirements of large distributed services.

1 INTRODUCTION

Web services are intended to provide for an efficient framework to ensure daily web transactions among partners who may own heterogeneous environments. Since web services are built in open distributed environment, the access control issues about web services become the key factor which restricts their use. Access control defends against illegal access by malicious attackers and honest users. Designing web service's access control appropriately is a difficult problem for two reasons. First, it is important that the access control decision takes the context into account to make the access control system suitable for dynamic web services environments. In addition, the number of web service requestors and their invocations, e.g e-voting web services, are becoming larger than ever. Therefore, it is essential to take the scalability problem into consideration. In this paper, we present how the agent technology can offer very interesting aspects that can be exploited to design web service's access control systems, which satisfy context-awareness and scalability requirements. We have applied this technology in implementing our TDRBAC (Trust and Dynamic Role Based Access Control) system using JADE (Java Agent DEvelopment) platform.

The remainder of this paper is organized as follows. The second section discusses previous works related to web service security. An overview of TDRBAC model is presented in section 3. In section 4, we analyze some agent and multi-agent characteristics that can be useful to fulfill the web service's access control model requirements by providing both context-awareness and scalability. Section 5 describes the proposed TDRBAC multi-agent based system. The TDRBAC implementation details, the clients requests simulation and the verification of the TDRBAC scalability are detailed in section 6. Finally, section 7 concludes the paper and describes the future work.

2 RELATED WORKS

Different models of access control have been proposed to deal with the efficient management of access rights. These models include DAC (Discretionary Access Control), MAC (Mandatory Access Control) and RBAC (Role Based Access Control) (Ferraiolo and Kuhn, 1992). Comparing with other access control models, the access permissions in RBAC model are not assigned directly to users but to roles which correspond to different job descriptions within an organization. RBAC model is widely used to facilitate the security administration through reducing the complexity of permissions management.

In web service environments, the subject which submits request and the object which offers service resource are dynamic. Therefore, access control models for web services must be able to adjust the access control strategy according to security-relevant contextual information. Several context-aware models (Bhatti et al., 2005; Li et al., 2009; Joshi et al., 2005) are extended the basic RBAC model. Notwithstanding the success of RBAC model and the extended proposed models, researchers have often found these models to be inadequate for open systems where the users' population is dynamic and the identities of all users are not known in advance. Several researchers have proposed credential-based access control models (Blaze et al., 1996; Li and Mitchell, 2003) where the user has to produce a predetermined set of credentials, such as credit card number to gain specific access privileges. Although, credential based models solve the problem of access control in open systems, they do not bind a user to its purported behavior or actions and they do not reveal whether the credential was obtained via deviated means. In (Bhatti et al., 2005; Chakraborty and Ray, 2006) the access control decision is based on the user trust level that can be determined by a set of parameters which constitutes the user profile. In our previous work (Jemel et al., 2010), we have proposed a novel access control model named TDRBAC. In TDRBAC model, the access decision depends on the user trust level determined based on the diverse session contextual information and on the occurrence of certain events during the opened session, which enhances the access control level.

Despite the contributions diversity, the majority of current solutions focuses only on the definition of access control models and rules. In fact, there are few works which are interested in access control systems design (Khemakhem et al., 2008; Wang and Wang, 2007) that can offer diverse interesting properties such as scalability (Calero et al., 2010; Ghali et al., 2010).

3 OVERVIEW OF TDRBAC MODEL

The definition of TDRBAC model is based on four main concepts :

- **Context-awareness**
  TDRBAC captures the relevant contextual information related to the user session environment which imposes some contextual constraints that should be incorporated in its access control decision. For example, in case of G2G (Government-to-Government) interactions and collaborations, an access request to an organization service coming from a machine with an IP address different from the IP addresses of the organizations which it interacts with can be denied.
- **Trust**
  In TDRBAC model the trust level is based on a set of contextual information. For example, an access request towards an organization outside of administration working hours can be considered as an untrustworthy request. The session context directly affects the user trust level, and hence the permissions granted to him/her during this session.
- **Event**
  TDRBAC model is also dynamic by providing a natural way to handle role access rights based on the occurrence of certain events. We define an event as any transient occurrence of a happening of interest such as the service term expiration or a completion of a process or a task.
- **Dynamic Role View**
  A view is a named set of access rights. Comparing with other access control models, the view assignment decision, in TDRBAC model, is not only based on the user credentials or its associated trust level, but also on the role already assigned to him/her, hence, the role view notion.

4 TDRBAC SYSTEM: REQUIREMENTS AND TECHNICAL SOLUTIONS

Agent-based system technology promises a new paradigm for conceptualizing and implementing software systems. In (Ferber, 1995), J. Ferber defines an agent as an autonomous entity, real or abstract that can act by itself, and its environment. In multi-agent systems, it can communicate with other agents and its behavior is the result of its observations, knowledge and interactions with other agents. An overview of the access control model requirements and the respective technical solutions offered by agent technology are detailed as follow.

4.1 TDRBAC Context Awareness

In (Singh and Conway, 2006), the agent is considered as an entity that is able to keep a model of the current context. It listens to the contextual information and communicates changes in its own context to other agents. Thanks to agent capabilities to perceive and respond instantly to changes in its environment, the software agents are used within many context-aware systems (Sahli, 2008). In order to incorporate the contextual information in TDRBAC system access control decision, we propose to define a *Context Collecting agent* which is in charge of listening, during the active session, to the environmental context information and to the different events that can occur. It communicates them to the others TDRBAC system components.

4.2 TDRBAC Scalability

In (Burness et al., 1999), scalability is considered as a measure of how the performance changes as the system size is increased. The most significant performance factor is the response time, as seen by a user under normal working conditions, coupled with the cost of the system - hardware requirements - per user. Multi-agent systems modularity offers an important solution to satisfy the software system scalability requirement. The term modular refers to the design of any system both hardware and software, made of separate modules that can be connected together. A multi-agent system is modular by default, since each agent can be seen as a single module. In addition to this, a single agent is composed of a set of different modules or layers. The multi-agent system modularity allows each agent to use the most appropriate paradigm for solving its particular problem and to share them with other agents.

To ensure the scalability of our TDRBAC system, we design it as a set of agents and we assign to each one of them some sets of tasks that can be accomplished. This decomposition ensures the treatment of several web service access requests in a simultaneous way which makes our system suitable for web applications that need to address a large number of user authorizations.

5 TDRBAC MULTI-AGENT BASED SYSTEM

5.1 Architecture Components

The architecture of the TDRBAC multi-agent based system is depicted in figure1. The client is the service requestor, SOAP interceptor agent and web services are the service providers. SOAP interceptor agent parses the SOAP message delivered by the requestor. It extracts the user's credentials and enforces the TDRBAC model which carries out permission assignment. TDRBAC system architecture includes the following agents :

- ***Role Assigning Agent***
  It is the TDRBAC component responsible for the role assignment to the web service requestor. The role assignment operation is based on a set of role assignment rules and it respects a set of constraints imposed by the system administrator.
- ***Context Collecting Agent***
  It accomplishes the function of collecting a variety of data related to the user session environment such as the access request time, the user location, etc. Moreover, *the Context Collecting agent* remains listening to every interesting event that can occur during the user active session. It can respond instantly to the different contextual changes and notifies the others TDRBAC components of them.
- ***Trust Evaluating Agent***
  The *Trust Evaluating agent* carries on evaluating the user trust level based the collected user session information.
- ***View Assigning Agent***
  It is designed with Prolog as the inference system to decide the appropriate view to be assigned to the user role. The *View Assigning agent* input is its knowledge base which is made up of the facts base and the rules base. The facts base maintains the role already assigned to the authenticated requestor, the user trust level evaluated by the *Trust Evaluating agent* and the events that can be occurred during the lifetime session. The rules base presents the different constraints imposed by the system administrator on the view assignment operation and a set of assignment permissions rules.

5.2 TDRBAC Architectural Components Interactions

The TDRBAC components interactions are presented as follow:

1. Initially, the SOAP interceptor agent intercepts the SOAP message sent by the web service user, extracts his credentials and sends them to the *Role Assigning agent*.
2. The *Role Assigning agent* checks the user credentials validity. If the user credentials are accepted the *Role Assigning agent* assigns the appropriate role to the web service requestor.
3. The *Role Assigning agent* informs the *Context Collecting agent* about the role al-

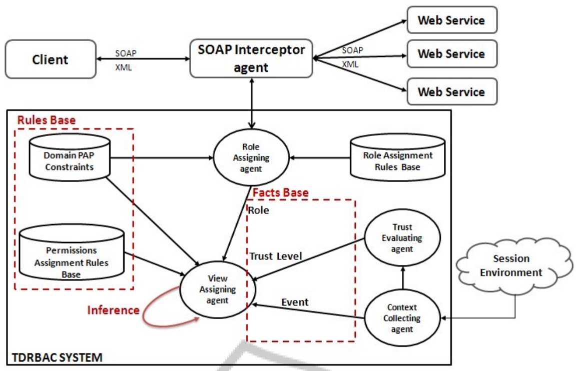

Figure 1: Architecture components of the TDRBAC multi-agent based system.

ready assigned to the user. Based on the user role, the *Context Collecting agent* collects the appropriate contextual information related to this session and it filters it. Then, it reports this information to the *Trust Evaluating agent*.

4. At the reception of the contextual information, the *Trust Evaluating agent* evaluates the user trust level and reports it to the *View Assigning agent*.
5. Based on the user role and the evaluated trust level, the *View Assigning agent* assigns to the user the appropriate view and the session is opened with the granted permissions.
6. During the active session, the *Context Collecting agent* remains listening to every context change and to the interesting events that occur. In the first case, it reports the new contextual information to the *Trust Evaluating agent* which evaluates the new user trust level and reports it to the *View Assigning agent* to change the user permissions . In the second case, it informs the *View Assigning agent* of the occurred events in order to change the granted permissions.

6 TDRBAC EXPERIMENTATIONS

Controlling the user access to the large-scale distributed web services should take the scalability problem into consideration. To verify the scalability of our proposed TDRBAC multi-agent based system, a prototype of our system is implemented and some experimentations relative to the implemented prototype are made.

6.1 Implementation Details

The TDRBAC multi-agent based system is implemented using JADE platform. In fact, JADE is implemented with Java language which enable programming in distributed heterogeneous environments. Moreover, it allows to build agent-based systems in compliance with FIPA specifications which allows the interoperability with agents acting on different agent platforms with the constraint that they must be FIPA-compliant.

We define 90000 different credentials associated to the authenticated clients and 4 roles that can be assigned to them. We attribute for each role 10 views. In addition, we define for each role 5 context attributes and their associated weights that can be incorporated in the TDRBAC access control decision.

6.2 Simulation and Performance Results

In order to analyze how our access control system scales with increasing numbers of simultaneous users, we run several performance tests. We use two different machines for our experimental setup. The first one is an Intel Core 2 Duo E7500 2.93GHZ with 2 Gbytes RAM running the TDRBAC system. The application that emulates the client requests is running in a Genuine Intel T2250 1.73GHZ with 2 Gbytes RAM. Both systems are interconnected by a crosser cable with 1-Gbit Ethernet link. To simulate a large-scale web traffic, it is important to model the web session arrivals, the session context change frequency and the session duration which affect the TDRBAC system performance.

- **Modeling Web Session Arrivals**
  The Poisson distribution $f(k;\lambda) = (\lambda\Delta t^k)/k!e^{-\lambda\Delta t}$, which gives probability of k arrivals to a system fed with an arrival rate λ over a period of length Δt, is a straightforward candidate for modeling session inflow to a web site. In fact, the Poisson distribution has been successfully used several times in performance analysis of systems with

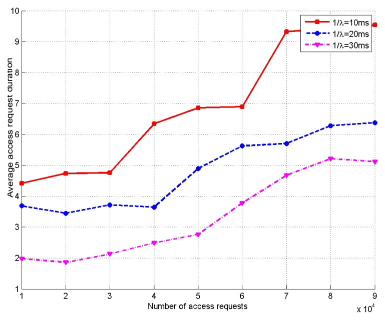

Figure 2: Average access request duration vs. number of access requests.

very large population of subscribers, e.g., a telephone central office, a node of packet-switched network and also an email server (Ohri and Chlebus, 2005; Darryl et al., 2003). For our simulation, we suppose that the web session arrivals are modeled by applying the Poisson process with parameter λ. As a result, the time between consecutive web session arrivals (called inter-arrival times) is an independent random variable exponentially distributed with parameter 1/λ.

- **Modeling Web Session Context Change Frequency**
  We suppose that, during the active session, all the instants have the same probability to be the instant of a session context change. Therefore, the uniform distribution with parameters *min=0* and *max=sessionDuration* is applied to model the session context change frequency during an active session.
- **Modeling Web Session Duration**
  The Pareto distribution is usually applied to model long session durations such as a FTP session (Goseva-Popstojanova et al., 2006). Ben Azzouna and al. demonstrate in (Azzouna et al., 2004), that the request duration can be approximated by a two-parameter Weibullian distribution $f(x;\eta;\beta) = (\eta/\beta)(x/\beta)^{\eta-1}e^{(x/\beta)^{\eta}}$ where η and β are, respectively, the scale and the skew parameters. For our simulation, we use an atomic web service i.e. the session duration is almost usually short, we apply the Weibullian distribution to model the web sessions duration.

In order to test the TDRBAC system performance, we stress the system with multiple requests for a long period of time. For the simulation process, we propose to define for each opened session a lifetime average equal to 60s, therefore, we fix η=60 and β=4 for the used Weibullian distribution. Moreover, we assume that, during the simulation process, the context of 20% of the opened sessions changes during

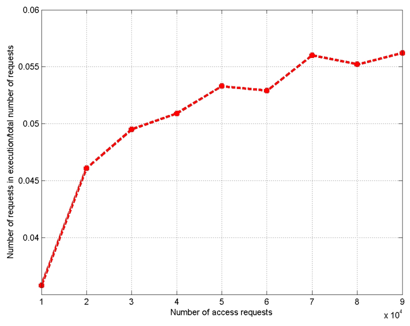

Figure 3: Number of requests in execution/total number of requests vs. number of access requests.

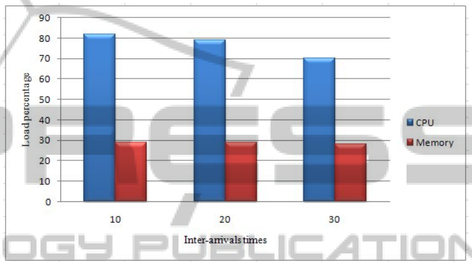

Figure 4: CPU and memory loads vs. inter-arrivals times.

their lifetime. For every single request, the processing duration is logged. A value of the average access request duration is calculated for every 1000 consecutive logged values. The figure 2 shows how the average access request duration varies with the total number of the arrived access requests in TDRBAC system for different session inter-arrival times (10ms, 20ms and 30ms). The results demonstrate the good performance of our system. For example, the step from 10000 requests to 90000 requests (9 times more) leads to an increase in the average access request duration from 4.41 to 9.53ms (only 2.16 times more) for the inter-arrival time 1/λ=10ms (100 requests/s). Moreover, it increases from 3.68 to 6.37ms (only 1.73 times more) for the inter-arrival time 1/λ=20ms (50 requests/s). In addition, by comparing the average access request duration for 90000 access requests for the inter-arrival time 1/λ=20ms with that for the inter-arrival time 1/λ=10ms (2 times more), it increases from 6.37 to 9.53ms (only 1.49 times more). The curve presented in figure 3 depicts the variation of the rate of the waiting requests in the TDRBAC system with the total number of launched requests. The waiting requests are the requests under treatment and have not yet received permissions. The results demonstrate that our system can handle a large number of requests. In fact, despite the increase in the total number of launched requests, the waiting requests rate is nearly constant. For example, the step from 30000 requests to 90000 requests (3 times more) leads to an increase in waiting requests rate from 0.049 to 0.056.(1.14 times more).

During every simulation process, we note both the CPU and the memory load percentage. As shown in figure 4, the CPU load percentage for the inter-arrival time 1/λ=30ms is 70% and it increases only to 82% for the inter-arrival time 1/λ=10ms. In addition, the memory load is nearly constant for the different simulations processes which show the good performance of our TDRBAC system.

7 CONCLUSIONS AND FUTURE WORK

In this work, we have presented an overview of our access control model called TDRBAC. This model is designed using the agent technology which satisfies context-awareness and scalability system requirements, and it is implemented using JADE platform. The performance tests of the prototype implementation show that the system is scalable for thousands of requests. In the future, we intend to stress the system with more requests and to duplicate the overloaded TDRBAC system agents. In addition, we plan to test our system with different system configurations such as roles number, views number, context attributes number, etc. Finally, we aim to test our system efficiency by applying different attack scenarios.

REFERENCES

Azzouna, N. B., Clérot, F., Fricker, C., and Guillemin, F. (2004). A flow-based approach to modeling adsl traffic on an ip backbone link. *Annales des Télécommunications*, 59(11-12):1260–1299.

Bhatti, R., Bertino, E., and Ghafoor, A. (2005). A trust-based context-aware access control model for web-services. *Distrib. Parallel Databases*, 18:83–105.

Blaze, M., Feigenbaum, J., and Lacy, J. (1996). Decentralized trust management. In *In Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173. IEEE Computer Society Press.

Burness, A.-L., Titmuss, R., Lebre, C., Brown, K., and Brookland, A. (1999). Scalability evaluation of a distributed agent system. *Distributed Systems Engineering*, 6(4):129.

Calero, J. M. A., Edwards, N., Kirschnick, J., Wilcock, L., and Wray, M. (2010). Toward a multi-tenancy authorization system for cloud services. *IEEE Security and Privacy*, 8:48–55.

Chakraborty, S. and Ray, I. (2006). Trustbac: integrating trust relationships into the rbac model for access control in open systems. In *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 49–58, New York, NY, USA. ACM.

Darryl, N. H., Veitch, D., and Abry, P. (2003). Cluster processes, a natural language for network traffic. In *IEEE Transactions on Networking*, pages 2229–2244.

Ferber, J. (1995). *Les Systmes multi-agents: Vers une intelligence collective*.

Ferraiolo, D. and Kuhn, R. (1992). Role-based access controls. *15th National Computer Security Conference*, pages 554 – 563.

Ghali, C., Chehab, A., and Kayssi, A. (2010). Catrac: Context-aware trust and role-based access control for composite web services. In *Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology*, CIT '10, pages 1085–1089, Washington, DC, USA. IEEE Computer Society.

Goseva-Popstojanova, K., Li, F., Wang, X., and Sangle, A. (2006). A contribution towards solving the web workload puzzle. In *Proceedings of the International Conference on Dependable Systems and Networks*, pages 505–516, Washington, DC, USA. IEEE Computer Society.

Jemel, M., Azzouna, N. B., and Ghedira, K. (2010). Towards a dynamic access control model for e-government web services. In *APSCC*, pages 433–440.

Joshi, J. B. D., Bertino, E., Latif, U., and Ghafoor, A. (2005). A generalized temporal role-based access control model. *IEEE Trans. on Knowl. and Data Eng.*, 17:4–23.

Khemakhem, M., BenAbdallah, H., and Belghith, A. (2008). Towards an agent based framework for the design of secure web services. In *Proceedings of the 2008 ACM workshop on Secure web services*, SWS '08, pages 81–86, New York, NY, USA. ACM.

Li, F., Wang, W., Ma, J., and Su, H. (2009). Action-based access control for web services. *International Symposium on Information Assurance and Security*, 2:637–642.

Li, N. and Mitchell, J. C. (2003). Datalog with constraints: A foundation for trust management languages. In *In PADL 03: Proceedings of the 5th International Symposium on Practical Aspects of Declarative Languages*, pages 58–73. Springer-Verlag.

Ohri, R. and Chlebus, E. (2005). Measurement-based e-mail traffic characterization. In *Proceedings of Performance Evaluation of Computer and Telecommunication Systems, SPECTS'05*.

Sahli, N. (2008). Survey: Agent-based middlewares for context awareness. In *Proceedings of the first International DiscCoTec Workshop on Context-aware Adaptation Mecanisms for Pervasive and Ubiquitous Services (CAMPUS)*, volume 11.

Singh, A. and Conway, M. (2006). Survey of context aware frameworks: Analysis and criticism. Technical report, The university of NORTH CAROLINA.

Wang, C.-D. and Wang, X.-F. (2007). Multi-agent based architecture of context aware systems. *Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*, pages 615–619.