I.J. Wireless and Microwave Technologies, **2012, 3, 62-68**
Published Online June 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815//ijwmt.2012.03.10

Available online at http://www.mecs-press.net/ijwmt

The Study of Access Control for Service-Oriented Computing in Internet of Things

Guoping Zhang[a], Jing Liu[b]

School of Computer & Communication Engineering, China University of Petroleum, Dong Ying, China

Abstract

In Internet of Things, computing and processing of information is the core supporting. In this paper, we introduce "Service-Oriented Computing" to solve the computing and processing of information in IoT. However, a key challenge in service-oriented environment is the design of effective access control schemas.We put forward a model of Workflow -oriented Attributed Based Access Control (WABAC), and an access control framework based on WABAC model. WABAC model grants and adapts permissions to subjects according to subject atttribute, resource attribute, environment attribute and current task, meeting access control request of SOC. Using the approach presented can effectively enhance the access control security for SOC applications, and prevent the abuse of subject permissions.

Index Terms: Internet of Things; Service-Oriented; Access Control; Workflow; Task; Attribute

1. Introduction

As we are moving towards the "Internet of Things" [1], millions of devices will be interconnected, providing and consuming information available on the network and cooperate. Because computing and processing of information is the core supporting of IoT, we introduce "Service-Oriented Computing" [2] to it where each device can offer its functionality as standard services [4]. As such, we can make shared resources available to one another through these devices in IoT. However, a key challenge in service-oriented environment is the design of effective access control schemas. In SOC, services will be invoked by a large number of temporary subjects, and at the same time authentication and authorization need to cross several security domains frequently.

In this paper, we present a Workflow-oriented Attributed Based Access Control model (WABAC) to address these issues. Section 2 introduces related work. Next section describes the basic concept and definition of WABAC model. In Section 4, we discuss the implementation of WABAC model. Section 5 gives an example. Finally, the conclusion is given and the future work is pointed out.

* Corresponding author.
E-mail address: [a]zhanggp@upc.edu.cn; [b]liujing415jsj@163.com

2. Related Work

Since the early 1960s and 1970s, it appeared several of access comtrol models, i.e. DAC, MAC and RBAC. Access control for distributed, heterogeneous service-oriented system is already becoming the hot topic in the field of network security. The model proposed by SHEN Haibo [5] can simplify the role assignment and management in heterogeneous system by using global roles and appropriate mapping to local permissions. Junqiang Zhu et al. put forward ACM4WSC model [6], by which the users can access the composite web service without concerning the access control of elementary services. Xu Feng et al. present a service-oriented role-based access control model and security architecture model for Web Services [7]. RBAC model will be faced with trivial role management work, and also can not solve large number of temporary users. Therefore, RBAC model is not suitable for access control of SOC.

In orded to realize resource sharing between coalitions, SPARTA ISSO first proposed Attributed Based Access Control (ABAC) in 2001 [8]. ABAC model defines permissions based on just about any security-relevant characteristics, known as attributes. It is more flexible and more powerful to describe complex, fine-grained access control semantics, which is especially suitable for the service-oriented architectures. Our paper puts forward Workflow-oriented Attributed Based Access Control model which has the following features:

- Introducing the concept of “Attribute”. Using attribute to describe detailedly entity’s properties (e.g., role may be an attribute of subject) and set access control policy flexibly.
- Using “Workflow control method”. Workflow is a business process which is composed of multiple relevant tasks. In WABAC, grant least permissions for subject according to attributes and current task, meeting “principle of least privilege”.
- Managing permissions dynamically. When a task is actived, subject has its own permissions. If a task is aborted abnormally, freeze subject’s permissions. Once a task is completed successfully, retrieve subject’s permissions.

3. Workflow-oriented Attributed Based Access Control Model(WABAC)

3.1. The concept of WABAC

1) Subject Attribute(SA): A subject is an entity that takes action on a resource, such as a user, application, or mobile devices. Each subject has associated attributes which define the identity and characteristics of the subject, such as subject’s identifier, IP address and Email address [9]. If there are K subject attributes, denote them by SA_k, k=1,2,…,K.

2) Resource Attribute(RA): A resource is an entity that is acted upon by a subject, such as service, data or smart device. As with subjects, resources have attributes (e.g., resource’s identifier, geographical position or creation date) that can be leveraged to make access control decisions. If there are M resource attributes, denote them by RA_m, m=1,2,…,M.

3) Environment Attribute(EA): Environment Attribute describes situational environment or context in which the information access occurs. Environment attributes such as current date, or the network’s security level, are different from subject attributes or resource attributes, but may be used in applying an access control policy. If there are N environment attributes, denote them by EA_n, n=1,2,…,N.

4) Attribute Distribution(ATTR): Attribute Distribution means attribute assignment for subject, resource and environment. ATTR(s), ATTR(r), and ATTR(e) are attribute assignment relations for subject s, resource r, and environment e, respectively:

$$ATTR(s) \subseteq SA_1 \times SA_2 \times \ldots \times SA_K$$

$ATTR(r) \subseteq RA_1 \times RA_2 \times \ldots \times RA_M$

$ATTR(e) \subseteq EA_1 \times EA_2 \times \ldots \times EA_N$

We also use the function notation for the value assignment of individual attributes. For example:

IP(s) = ”101.126.1.15”

ServiceName(r) = ”TVOnline”

CurrentTime(e) = ”09:30”

5) Task(T): Task is a fundamental unit of business work or business activity. And it is distinguishable action which may be relevant to multiple users or include several subtasks [10]. For example, Electronic Toll Collection flow includes three tasks: information collection, vehicle billing and bank charges.

6) Task State(TS): No less than any course, task has also its own lifecycle in which including five states, i.e. reday, active, hold, end and invaild that showen in Fig. 1.

7) Authorization Unit(AU): Authorization Unit is an abstraction of task, being made up of access subject, O and access permissions, P. If AU need external user to participate in, see the external user as access subject, O. If not, set O as null. P is the least set of permissions that complete a task. According to the executing process of workflow, AU can be divided into four categories: order unit, select unit, loop unit and concurrent unit.

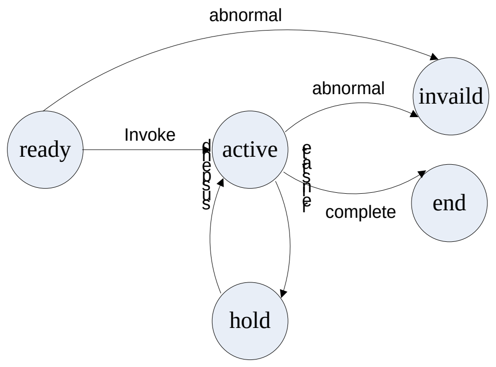

Fig 1. The State Transition of Task .

8) Authorization Dependency (AD): Authorization Dependency is the relations between AU in workflow. For any AU_1 and AU_2, the main dependency includes the following:

a) Order Dependency: AU_2 can be activated only after AU_1 has been finished, being written as $\mathbf{AU_1} \longrightarrow \mathbf{AU_2}$.

b) Defeat Dependency: AU_2 can be activated only after AU_1 has defeated, being written as $\mathbf{AU_1} \not\longrightarrow \mathbf{AU_2}$.

c) Divided Permission Dependency: AU_1 and AU_2 must be executed by different user, being written as $\mathbf{AU_1} \longleftrightarrow \mathbf{AU_2}$.

d) Agent Dependency: AU_1’s permissions can be surrogated to AU_2 when AU_1 is aborted, being written as $\mathbf{AU_1} \overset{d}{\not\longrightarrow} \mathbf{AU_2}$.

3.2. WABAC Model

WABAC = { S, R, E, P, T, AU } where: S is subject; R is resource; E is current environment; P is the set of permissions; T is the set of tasks; AU is the set of authorization unit. This model includes the following relations:

1) Permission Assignment: a many-to-many permission to subject assignment relation. Permission assignment depends on subject attribute, resource attribute, environment attribute, and current task.

2) Authorization Dependency: Authorization Dependency decides the execution flow of workflow. The relationship between AU is $AU \times AU \subseteq 2^D$, D={Order Dependency, Defeat Dependency, Divided Permission Dependency, Agent Dependency}.

3) Permission Transformation: In the process of access, subject's permissions change with task. Here we use authorization unit sequence to show permission transformation, such as {AU_1, AU_2, …, AU_n }, AU_1, AU_2, …, AU_n are specific authorization unit (Fig. 2 depicts the access control view of WABAC).

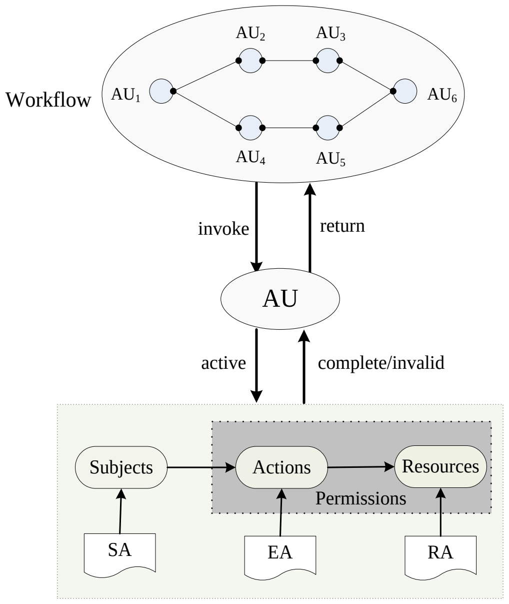

Fig 2. The Access Control View of WABAC.

4. Implementation of WABAC

4.1. Presentation of Attribute

SAML (Security Assertion Markup Language) which is published by OASIS is an XML-based standard for web service security, and provides a standards-based approach to the exchange of information, including attributes, that are not easily conveyed using other WS-Security token formats [12]. SAML defines three declarations: authentication declaration, attribute declaration and authorization decision declaration which carries authentication information, attribute information and authorization decision information respectively. In addition, SAML bindings defines the means by which lower-level communication or messaging protocols (such as HTTP or SOAP) are used to transport SAML protocol messages between participants. A simple example about SAML attribute declaration can be seen in Fig. 3.

```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
              AssertionID="5etf362baa7"
              Issuer="http://www.quadrasis.com/easi"
              IssueInstant="2010-12-20T08:30:25Z">
    <saml:AttributeStatement>
        <saml:Subject>
            <saml:NameIdentifier> afend </saml:NameIdentifier>
        </saml:Subject>
        <saml:Attribute
                  AttributeName="occupation"
                  AttributeNamespace="http://www.quadrasis.com/easi">
            <saml:AttributeValue>
                 Taxi driver
            </saml:AttributeValue>
        </saml:Attribute>
    </saml:AttributeStatement>
</saml:Assertion>
```

Fig 3. A Simple Example about SAML Attribute Declaration .

4.2. *Access Policy Language*

In 2003, OASIS put forward XML Access Control Markup Language named XACML which provides support for attributed based access control [13]. XACML is an XML-based language, which can describe access control policy, and provide a series of logical algorithm to control all of authorization process. Compared with other policy language, XACML access policy is based on subject attribute, resource attribute and environment attribute, and can achieve fine-grained access control. In addition, it can define new functions, data structure and combinational logic algorithm as needed.

This paper proposes an extension to the XACML architecture that showen in Fig. 4 based on SAML and XACML.

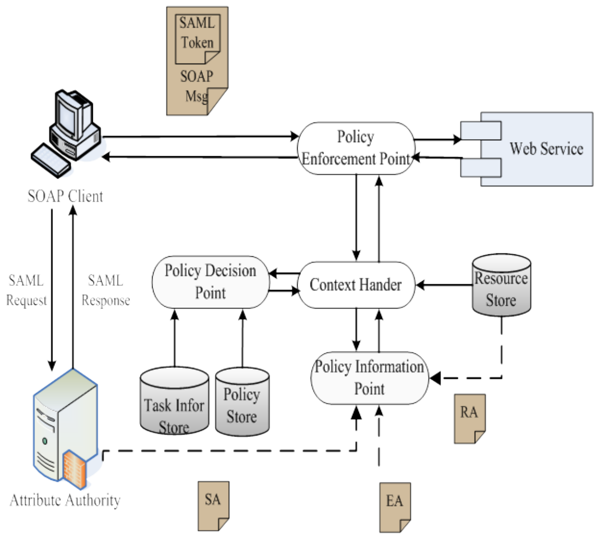

Fig 4. WABAC Access Control Framework .

- Before subject accesses system, subject must have SAML attribute declaration which is published by Attribute Authority. Then insert SAML declaration into SOAP Head, and send SOAP request to system access port via SOAP client.
- When system gets subject's access request, generate corresponding task instances, allowing each task to be in a ready state. As for these tasks, they will be carried out automatically in definite logical order.
- Once certain task is activated, Policy Enforcement Point (PEP) shoule pick up subject attributes and current task information from SOAP request message. At the same time, write current task state into Task Information Store, and create a standard XACML authorization request, send it to Policy Decision Point (PDP).
- After receiving authorization request, PDP will make authorization decision in the light of corresponding policies from Policy Store and current task state from Task Information Store. If need other attributes in authorization decision, get them through Policy Information Point (PIP).
- Context Handler gets the authorization decision, and transforms it into a format which can be accepted by PEP. Then PEP enforces the authorization decision. When certain task is completed successfully, start next task at once.

5. an example

Take Electronic Toll Collection system as an example of WABAC application. Electronic Toll Collection system includes information collection service, vehicle billing service and bank charges service. The system workflow is following:

- When vehicle is close to toll station, send access request to system. The system creates information collection task, vehicle billing task and bank charges task that will be carried out automatically according to information collection->vehicle billing->bank charges.
- Information collection equipments at side of road take pictures of vehicle, read electronic tag which is installed in vehicle, and get the only identification code of vehicle and other information that will be sent to control centre.
- Control centre looks for vehicle attributes (e.g., vehicle type, the owner name) from vehicle information store depending on the only identification code of vehicle, which is used to judge whether the vehicle is charged on. If need to charge, send request to vehicle billing service, and the service returns billing result to control centre.
- After sending the owner account information and billing result to bank charges service, this service will deduct certain cost from the owner account, add to vehicle toll account, and return charge result to control centre.
- Finally, send charge result to vehicle terminal, if bank charges task is completed successfully, show "Success Charge". The system will send "permission" command to the exit intercept equipment, allowing the vehicle to pass through the exit.

6. conclusion

In this paper, we focus on the access control of Service -Oriented Computing in Internet of Things. We present a workflow-oriented attributed based access control model and access control framework for SOC. Compared with other models, this model can achieve fine-grained access control, and manage permissions dynamically by introducing the notion of Attribute and Task, supportting principle of least privilege and separation of duties principle. It is very suit for access control of the service-oriented architectures, especially workflow based distributed computing system. In the future, we will use Apache Axis2 platform to deploy this model, and prove its validity.

References

[1] International Telecommunication Union UIT, “ITU Internet Reports 2005:The Internet of Things”, 2005.
[2] Papazoglou M.P, “Service-oriented computing: Concepts, Characteristics and directions”, In: Proceedings of the 4th International Conference on Web Information Systems Engineering, 2003.
[3] W3C Working Group Note, “Web Services Architecture”, l1 February 2004.
[4] Patrik Spiess and Stamatis Karnouskos, “SOA-based Integration of the Internet of Things in Enterprise Services”, 2009 IEEE International Conference on Web Services.
[5] SHEN Haibo and HONG Fan, “A Context-Aware Role-Based Access Control Model for Web Services”, Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE 05).
[6] Junqiang Zhu, Yu Zhou and Weiqin Tong, “Access Control on the Composition of Web Services”,Proceedings of the International Conference on Next Generation Web Services Practices(NWeSP 06), 2006.
[7] Xu Feng, Lin Guoyuan, Huang Hao, and Xie Li, “Role-based Access Control System for Web Services”, Proceedings of the Fourth International Conference on Computer and Information Technology (CIT 04).
[8] http://www.isso.sparta.com/documents/
[9] Eric Yuan and Jin Tong, “Attributed Based Access Control (ABAC) for Web Services”, Proceedings of the IEEE International Conferenceon Web Services (ICWS 05), 2005.7:560-569.
[10] R.K.Thomas and R.S.Sandhu, “Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management”, Proceedings of the IFIP WG11.3 Workshop on Database Security, Auguest 1997.
[11] Xiangning Zhou and Zhaolong Wan, “An Access Control Model of Workflow System Integrating RBAC and TBAC”, In IFIP International Federation for Information Processing, vol. 252, 2007, pp.246-251.
[12] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
[13] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
Torsten Priebe, Wolfgang Dobmeier, Christian Schläger, and Nora Kamprath, “Supporting Attribute-based Access Control in Authorization and Authentication Infrastructures with Ontologies”, Proceedings of the First International Conference on Availability,Reliability and Security (ARES 06), April 2006.