2012 IACSIT Hong Kong Conferences
IPCSIT vol. 29 (2012) © (2012) IACSIT Press, Singapore

Role and Data-Based Constraints of Data Access Control in a Legacy System Migration to a Service-Oriented Environment

Richard Millham[1+], Evans Dogbe[2], Prenitha Singh[2]

[1] Durban University of Technology/University of Bahamas, Nassau, Bahamas/Durban, South Africa
[2] Durban University of Technology, Durban, South Africa

Abstract. Migration of legacy systems to a service-oriented environment brings a host of security migration challenges. Certain legacy systems were often built to function in a secure networked environment, which were governed by coarse granular access lists and dispersed security mechanism and which relied on poorly-defined roles. These roles, with their coarse granular data access, may have functioned adequately in a restricted network environment but after their migration to a service-oriented environment, where system functionalities operate as independent Web services/clients and where these services face greater security threats, these existing roles and their subsequent access rules are no longer sufficient. Before new service-oriented security mechanisms can be implemented, the legacy system use cases, along with their associated roles, operations and access rules, must be reconsidered and reworked. In this research-in-progress paper, we propose an improved service-oriented model to allow use cases with multiple roles per user, as is often needed in a business environment. Furthermore, this model must depict rules, which take into account a specific role, operation, and data view associated with a user's action, in order to determine whether the action should proceed and, in so doing, reduce some of the security vulnerabilities of a service-oriented environment.

Keywords: legacy system migration, service-oriented security, data access control

1. Introduction

The migration of a legacy system to a service oriented environment is a multi-faceted effort [10]. Legacy systems must be analysed and then their functionalities must be identified as features, which are then logically grouped into services, which are then migrated to Web services/clients [9]. Legacy data must frequently be transformed to a relational model that is usable by service-oriented architecture [8]. One migration aspect that has been frequently ignored is the migration of legacy system security to a service-oriented environment. Although legacy systems may differ greatly in terms of architecture, domain, and programming language [15], many legacy systems often functioned as a monolithic, standalone system or within a secure networked environment; migrating these systems to a less secure networked environment requires a reconsideration of these systems' security mechanisms and, before these mechanisms can be reconsidered, a fundamental rethinking of the system's users, roles, operations, and data access along with the interactions amongst them.

An example, consider a select legacy system with user permissions based solely on access control lists. In this system, a user would log in and their log in would indicate their role and, based on the access control list, a list of permittable accessible resources would be provided. As logins and roles were often combined, one login would usually give the maximum access to resources as stipulated by the top role associated with that login ("maximum privilege") regardless of the role currently being used. Because of the "maximum privilege" rule, maximum database access was granted which exposed greater security vulnerabilities for that

+ Corresponding author. Tel.: + 1(305)897-2154;
E-mail address: richardmillham@hotmail.com.

resource. An example, a user logging in as a payroll clerk would be granted access to all payroll records with full control, rather than the employees under their purview. This grant would allow a hacker, once he gained a login as payroll clerk, or disgruntled payroll clerk to use their access to change the salaries of all employees that they are granted access to or to view the salaries of people outside their role.

In order to avoid this problem, this legacy system policy of granting wide access based on users must be rethought and a new system of security access must be produced. This new system considers many facets of security when performing the data access: the role and operation being performed (role and context), the group of data that is being affected (is it part of or outside the prescribed operation?), and if a role is required which requires greater privilege, the system should request further authorisation. In this paper, we present part of a model for revised security for legacy systems. This model provides a way for the migrating developer to specify a finer granular data access than was previously permitted under the legacy access control list as well as being able to extend role and context-based data control.

In the organisation of this paper, we briefly survey related work in security models and modelling with their advantages and disadvantages. We propose a model for data access that is constrained by a specific role, operation, and group of datarows associated with people being managed under that role. We propose a few methods to semi-automate the gathering of data necessary for the formulation of constraints which form our rules.

2. Related Work

Different security models enable users to express their security needs and their associated threats and vulnerabilities while accessing their impact. Sandu [13] presents a unified model of role-based access, both flat and hierarchical, with permissions dependent on roles. However, this model fails to take into account specific actions associated with a role and the permissions that need to be associated with an action.

Wada extends UML to encompass service-oriented architectures but acknowledges that UML has difficulty modelling and expressing access policies. His UML-based model uses a message stereotype to model a security token, modelled as a message, which is passed to Web services to allow them to respond to requests. The creation of these security tokens, based on access policies along with their access definitions, must be performed by the developer.[14] EBIOS (Expression of Needs and Identification of Security Objectives) provides a consistent risk analysis within its model but neglects the communication infrastructure that is a crucial part of service-oriented architecture. Another model, OCTAVE (Operationally Critical Threat, Analysis, and Vulnerability Evaluation) provides a strong risk and impact analysis with emphasis on the system architecture. This model evaluates the level of risk and possible impact emanating from that risk. In addition, this model analysis the access control of data required by a service: by role and by operation. Control over this data access is managed by a means of an access control list which controls the range of operations by user, rather than the more traditional allowable resources by user [11].

The basic modelling notation of Business Process Modelling Notation (BPMN), which forms the core of much system modelling, has been extended by Rodriguez to include security information. However, this information is focused on security authorisation with little attention paid to other security aspects [12]. Although out of the scope of this paper, Menzel argues that modelling should just not only include abstract security intentions but should contain additional security meta information which includes information of the data's value, trustworthiness, and dependencies between modelled entities. This additional meta information on data is often required because this data is passed through multiple intermediaries, including Web services, before it is processed or stored. Each of these intermediaries often requires this information to be able to function securely and to pass this data on to another intermediary [6].

Basin extended UML to form SecureUML which provides a means to specify role-based control and constraints. However, SecureUML does not seem to be able to provide fine-grained access control [2]. Data access control, in the form of coarse-grained constraints, is modelled by SecureSOA which incorporates them into a security policy to be implemented in WS Security mechanisms. SecureSOA is a security modelling language that is integrated with a system structure modelling language, FMC, which models the constraints and interactions between web services and lower level data access mechanisms. SecureSOA uses

a generic object template to indicate a web service or client, with accompanying information, that interacts and exchanges information with other objects, which are governed by various security intentions such as user authentication, identity provisioning, and data confidentiality [6]. However, SecureUML does not seem to provide adequate fine-grained protection for its data or a mechanism to incorporate roles and tasks within its security meta-model.

Service-oriented environments require several mechanisms to ensure security for their systems. WS Security policies have encoded security policies which govern the interaction between different participants in a service oriented environment [2]. One of these needed mechanisms is secure message transmission in a non-trusted network. WS-Security specifications can provide secure message transmission between service clients and providers through encrypted message and digital signatures for non-repudiation. However, these specifications can supply only basic identity authentication and they are unable to provide fine-grained data protection. Even though users may be authorised, any service oriented access policy must take into account the role, operation, and data view in order to prevent a SQL injection attack [16]. As the user role changes from one domain to another more secure domain, different authentication mechanisms should be used to authenticate the user before they can gain access to a more secure domain [4]

3. Our Model for Rule-Based Constraints

In order to reconsider roles, operations, and data access for a new service environment, we must first examine the legacy system's use cases, determine their data dependencies, and reformulate them into rule-based scenarios modelled as constraints in our new model.

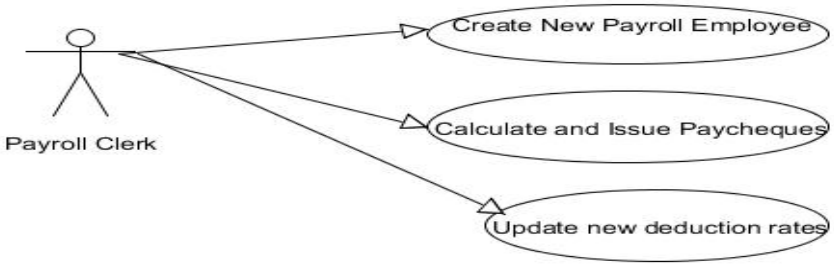

Fig 1. Legacy payroll use cases

Consider the selected legacy system's use case for payroll. A payroll clerk performs the payroll with many activities, such as create new payroll employee. Typically, the same login and role would suffice for all activities, even though from a security standpoint, each of these activities has a different role, operations, and data accesses. The activities and the data accesses that are associated with each use case are as follows:

- Create new employees for payroll – which involves write/append access to the employee pay rate table
- Calculate and issue paycheques – which involves read-only access to the employee pay rate table and deductions table with write access to the Accounts Receivable table
- Update new deduction rates – which involves write access to the deductions table

All of these operations contained in the subdivided use cases can be further limited to the subset of employees under the clerk's responsibility and to subset of the accounts receivable table that is limited to payroll entries. Hence, there are multiple restrictions on data access: firstly, on the basis of roles, and secondarily, on the basis of specific operations.

Hence, the basis for this restriction is User (U) with Role (Ri) has a subset of managed people (Pa..Pn) under their purview which interact with a specific operation (Oi) which, in turn, controls a set of data accesses (Di..Dg). The rule states that only data accesses that are a subset of the allowable data accesses (DA) for operation Oi are permitted.

The business rule notation selected to express these conditions is that of transition graphs as the processes highlighted in the use case often are comparable to transitions from state to state [5]. Hence, the business rule emanating from the role to the interaction in the meta-model would be (User Authorised) AND

(Ri ⊂U :(Ra..Rn)) AND (Oi⊂Ri (Oa...On) which means that the user must be authorised and the user's selected role must within a subset of roles defined under that user. The business rule emanating from the operation to the interaction in the meta model would be (Oi⊂Ri(Oa,,On) AND (DA⊂Oi(DAa,,DAn) which means that the operation must be within a subset of operations defined for the selected role and that the requested data access must be within the subset of data accesses defined for that selected operation. Both of these two conditions must be fulfilled before authorisation can occur. A further requirement for this business rule could be Ps⊂Ri (Pa...Pn) AND DAi⊂T: Ps (Rwa...Rwn) which means that a people subset, Ps, must be a subset of people under the purview of the selected role, Ri, and that the requested data access, DAi must interact only with the rows (Rw) of database tables that are associated with this people subset, such as payroll.

Figure 2 highlights a partial meta-model that illustrates the rules embedded in the new modelling. Each user is modelled, each with one or more roles. Each role has a list of permitted operations attached to it. Each role is associated with one or more operations, which can consist of one or more activities. Each operation has a set of allowable data accesses. An interaction determines whether a data access requested by an activity is within the set of permissible data accesses for the enclosing operation as well as the selected operation is within the set of allowable operations for that role. If the determination is that both the operation per the specific role and data access(es) per the specific operation are allowed plus the data accesses are performed on data rows associated with a subset of people listed as being managed under that role, authorisation is given for input/output and the data access proceeds. If, however, the data access is outside the set of permissible accesses for that operation or the operation is outside the role or the data accesses are on data rows that are not associated with the list of people being managed under that role, authorisation is denied and no input/output occurs .In the latter case, if a user requires a role with a greater privilege (such as updating employee salaries rather than reading them), multi-tiered security architecture would demand further authentication before the user is allowed to proceed [16].

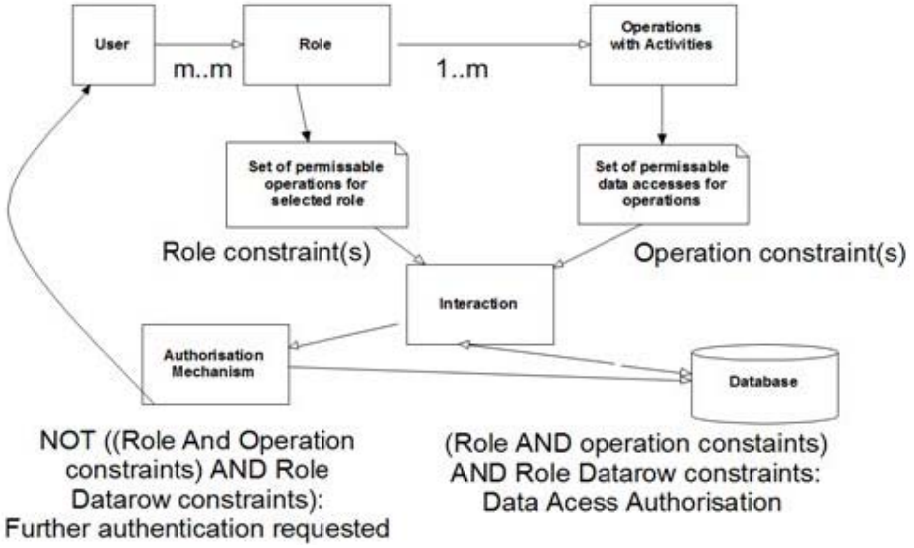

Fig 2. Meta-model of data access

This meta–model addresses a crucial security weakness in many systems: SQL injection attacks. A hacker may adopt a role, such as payroll cheque issuance, and then through SQL injection of input cause the affected table to perform unusual actions – such as updating the payroll rate of employees rather than simply reading them. Since the legacy system access control list may have granted access to this table to anyone with a role of payroll cheque issuance in order to access the data needed for this operation, this data update would proceed. The meta-model provides protection on several fronts. If a user, with the role of payroll cheque issuance, tries to update the employees' pay rate, this data access will be disallowed as it is not part of an operation associated with that role and it is not part of the set of data accesses associated with any operation attached to that role. Adopting another role with their associated operation of updating employee payroll rates would cause the system to request further authentication from the user before allowing the user

to proceed. In addition, even if the role of updating employee payroll data was allowed for that user, the user would only be allowed to update the employee data for employees under their purview, rather than all employees in that table.

4. Implementations to Assist in the Formulation of Constraints

In order to implement this restrictive security policy, several steps need to be taken to gather information to help the formulation of these constraints which form rules. One action is to identify the particular data rows associated with an operation in order to model these rows as a permitted subset of data. In order to identify these rows, it is necessary to perform a dynamic trace of calls to the legacy data system as the operation is performed along with dataflow and dependency analysis ([1]; [7]; [3]). This analysis will often produce sets of data that this legacy operation will involve in a particular scenario. In addition, static code analysis and expert guidance is often needed to find further sets of data that are involved in exceptional or missed cases in this same scenario. The purpose of this data analysis is to determine the sets of data that are associated with a specific operation tied in with a particular role and, thus, eliminate some of the tedious work in identifying fine-grained bits of data as being associated with this specific operation.

In addition to associate data with role-operations, another task that can be semi-automated is the association of people that are being managed by a specific role. In Fig. 2, one of the constraints was that the data access request must relate to specific rows of tables that are associated with the pool of managed people under a particular role. In other words, if a pension clerk is responsible for issuing the checks of Tom, Dick, and Terri, only the rows of the tables associated with these people can be accessed, and no other. This rule prevents a hacker, or disgruntled employee, from viewing the records of people other than whom they manage. In order to implement these constraints, a set of correspondence between the records of tables and a list of people being managed under a particular role is performed to ensure that only those records associated with this particular list is allowed data access. In order to establish this correspondence, a mapping of tables with their fields to an identifying attribute of a person within the list is needed. An example, the social security number field of the person must match with the social security number field in the list of people under that particular role.

5. Small Example

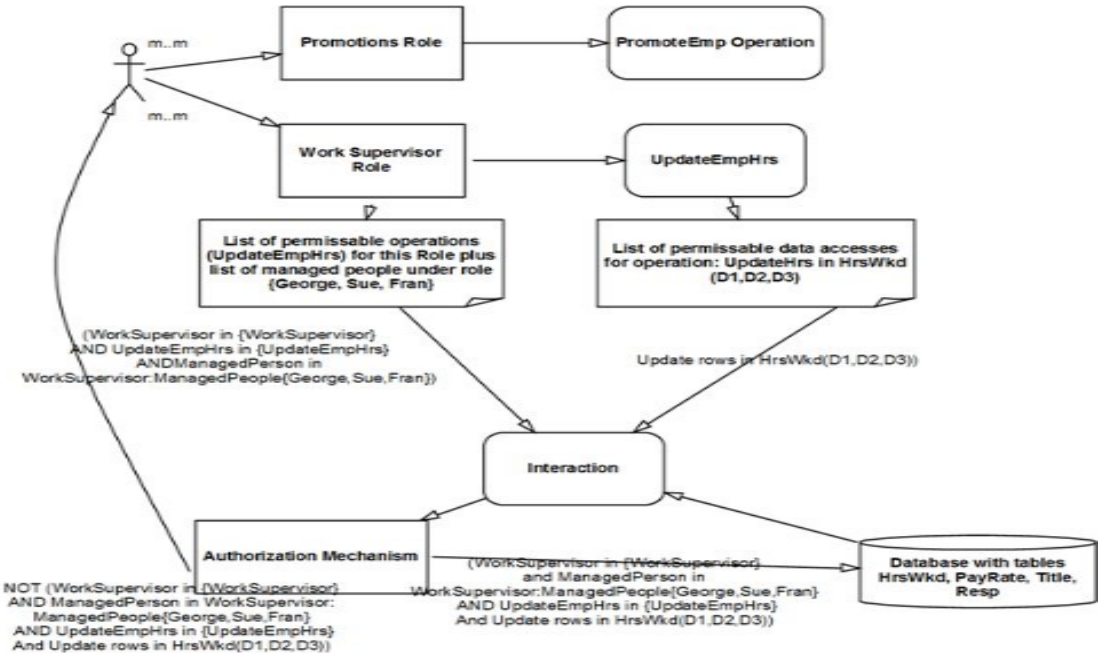

Fig 3: Example of our security model

A small example is given to illustrate our model and methods. A supervisor, Sam, manages three people. These people, with their employee ids, are {Sue, 100}, {George, 101}, and {Fran, 102}. Same has two roles: WorkSupervisor and Promotions. Each of these roles has a set of operations associated with them. The

WorkSupervisor has an operation, UpdateEmpHrs, which involves the updating the affected data rows of the HrsWkd table. Through our method of mapping managed people under roles to their associated data rows using some common characteristic such as employee id, it is determined that the UpdateEmpHrs will involve only data rows D_1 to D_3 in the HrsWkd table. The Promotions role has an operation, PromoteEmp, which involves updates to data rows D_1 to D_3 in the PayRate, Title, and Resp tables. The permissions and actions for this role are not shown for simplicity sake.

If Sam logins in and selects the role, WorkSupervisor, and then performs the operation, UpdateEmpHrs, on employees Sue and George, the data updates to table HrsWkd will be authorised and succeed with success output returned to the Interaction. If the UpdateEmpHrs operation tries to access the Payrate table, which is not a permissible data access under its allowable data access listing, or if Sam tries to update an employee, Fred, not listed under his Role's list of employees, the data update will not be authorised. In the case of a failed data access authorisation, such as attempting to update the Payrate table under the incorrect role, further authentication is required.

6. Conclusion

We presented a model that depicts constraints as per role and as per operation in order to restrict unwarranted data accesses that create security vulnerabilities, notably SQL injection attacks. This model demands a finer-grained definition of roles, operations, and their allowable data accesses than the more loosely defined and implemented security model of legacy systems which often operated in a trusted network environment. This finer grain of control model is more suited to the service-oriented environment where more security vulnerabilities exist and this reworking of the legacy system's security to a new finer grained model is a necessary first step, before security implementation can occur, in the security migration of the legacy system to the security oriented environment.

7. Future Work

Future work might include developing mechanisms to implement our security model and to semi-automate the development of constraint formulation. Besides focusing on authorisation, future work might include integrating other aspects of service-oriented security in a centralised security service that governs the now dispersed Web services that once formed the legacy system.

8. References

[1] Bianchi, A., Caivano, D., & Visaggio, G. (2000). Method and Process for Iterative Reengineering of Data in a Legacy System, WCRE, Los Alamos: IEEE Press.

[2] Basin, D., Jurgen Doser, and Torsten Lodderstedt "Model driven security: from uml models to access control infrastructures". ACM Transactions on Software Engineering and Methodology,15(1):39{91, January 2006.

[3] Cleve, A., Henrard, J., & Hainaut, J.-L. Data Reverse Engineering using System Dependency Graphs, WCRE. Los Alamos: IEEE Press, 2006.

[4] Fumiko, Satoh "Adding Authentication to Model Driven Security", ICWS, 2006.

[5] Herbst, H,. G. Knolmayer, T. Myrach and M. Schlesinger The Specification of Business Rules: A Comparison of Selected Methodologies , IFIP, 1994

[6] Menzel, M. "SecureSOA – Modelling Security Requirements for Service-Oriented Architectures", IEEE Conf. On Services Computing, 2010.

[7] Millham, R. "Evolution of Batch-Oriented COBOL Systems into Object-Oriented Systems through Unified Modelling Language", Unpublished doctoral dissertation, De Montfort University, Leicester, UK, 2005.

[8] Millham, R., H. Yang Industrial Report: Data Reengineering of COBOL Sequential Legacy Systems". COMPSAC, Los Alamos: IEEE Press, 2009.

[9] Millham, R. "Migration of a Legacy Procedural System to Service-Oriented Computing Using Feature Analysis", *ECDS-CISIS*, Cracow, Poland, 2010.

[10] Millham, R "Software Asset Re-use: Migration of Data-Intensive Legacy System the Cloud Computing Paradigm"

in Software Reuse in the Emerging Cloud Computing Era, IGI Group, USA (in press), 2011

[11] Ouedraogo, W., F. Bennier, N. Salatage “Security Preference Specifications in a Service-Based Workflow”, Sixth Intl Conf. On Information Assurance and Security, 2010.

[12] Rodrıguez, A. E. Fern´andez-Medina, and M. Piattini, “A bpmn extension for the modeling of security requirements in business processes,” IEICE Transactions, vol. 90-D, no. 4, pp. 745–752, 2007.

[13] Sandhu, R., D. Ferraiolo, R. Kuhn “The NIST Model for Role-Based Access Control: Towards a Unified Standard”, ACM Workshop on Access Control, 1992.

[14] Wada. H A Service-Oriented Design Framework for Secure Network Applications, Compsac, 2006.

[15] Weber, Carl (2006) Assessing Security Risk In Legacy Systems”, Cigital, Inc, Retrieved Aug 26, 2010 from https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/legacy/624-BSI.html. 2006

[16] Zhao, F., Xin Peng, Wenyun Zhao “Multi-Tier Security Feature Modeling for Service-Oriented Application Integration”, Eighth IEEE ACIS International Conference on Computer and Information Science, 2009