韓國컴퓨터情報學會 論文誌
第15卷 第1號, 2010. 1.

2010-15-1-4-3

OS-RBAC과 임무분리 정책의 통합 관리 모델

변 창 우*

An Integrated Management Model of OS-RBAC and Separation Of Duty Policy

Chang-Woo Byun*

요 약

임무분리(Separation of Duty: 이하 SoD)는 위임(delegation)과 더불어 접근제어 분야에서 중요한 보안 정책 중의 하나이다. 기존의 역할 기반 접근제어 모델을 기반으로 한 과거 임무분리 모델은 역할기반 접근제어 모델의 여러 제한적인 구성 요소(역할계층, 역할계승 등)에 의해 조직 내에서 관리하는 정보의 무결성 침해 가능성에 대한 최소화를 충분히 해결하지 못하고 있다. 본 논문에서는 관리적 역할기반 접근제어 모델인 OS-RBAC 모델과 임무분리 정책을 통합한 조직 구조 기반의 임무분리 관리 모델(OS-SoDAM)을 제시한다. OS-SoDAM은 분산 기업 환경의 관리 행위의 범위 기준을 역할 계층과는 구별된 조직 구조로 정의하고 있어 역할계층 및 역할계승에 제한적인 임무분리의 한계를 극복하고 자유롭게 임무분리 정책을 접근제어 모델에 적용할 수 있는 유연성을 제공한다.

Abstract

Like most large organizations, there are business rules such as ‘separation of duty’ and ‘delegation’ which should be considered in access control. From a SOD point of view, previous SOD models built on the (Administrative) Role-Based Access Control model cannot present the best solution to security problems such as information integrity by the limited constituent units such as role hierarchy and role inheritance. Thus, we propose a new integrated management model of administration role-based access control model and SOD policy, which is called the OS-SoDAM. The OS-SoDAM defines the authority range in an organizational structure that is separated from role hierarchy and supports a decentralized security officer-level SOD policy in which a local security officer can freely perform SOD policies within a security officer’s authority range without the security officer’s intervention.

▸ Keyword : 정보보호(Security), 접근제어(Access Control), 임무분리(Separation of Duty), 역할기반 접근제어 (Role Based Access Control), 조직 구조 기반 역할기반 접근제어(Organizational Structure- RBAC)

• 제1저자 : 변창우
• 투고일 : 2010. 01. 12, 심사일 : 2010. 01. 18, 게재확정일 : 2010. 01. 26.
* 인하공업전문대학 컴퓨터시스템과 교수
※ 이 논문는 2008학년도 인하공업전문대학 교내연구비지원에 의하여 연구되었음.

Ⅰ. 서 론

임무분리(separation of duty) 정책은 사용자들의 권한 남용이나 오용에 의한 사기, 공모를 방지하기 위하여 민감한 권한들은 한 사용자에게 모든 것을 부여하지 않고 여러 사용자에게 분산하여 부여해야 한다는 보안 원리이다[1, 2]. 이와 같이 민감한 권한들의 관계를 동•정적으로 구분하여 지정하고 취소하는 행위와 구별된 권한을 사용자에게 할당하고 회수하는 행위는 보안 관리자의 권한이기 때문에 관리적 접근제어 모델이 필요하다. 즉, 적용할 임무분리 정책이 보안관리자의 관리 영역을 벗어나지 않도록 하는 관리적 접근제어 모델이 필요하다.

과거 역할기반 접근제어(Role-Based Access Control, 이하 RBAC) 모델을 기반으로 한 관리적 역할기반 접근제어(Administrative RBAC, 이하 ARBAC) 모델 상의 임무분리 모델들은 임무분리를 제약조건(constraint)의 하나로 다루며 역할 수준 혹은 인가권한(permission) 수준의 임무분리 지정을 지원한다. 이러한 연구들은 구현과 실용성의 관점에서 보면 보안 관리자의 관리 영역 기준을 역할 계층으로 하였기 때문에 역할 계층에 의해 발생되는 문제점을 그대로 안고 있다.

본 논문은 다수의 보안 관리자가 필요한 분산 접근제어 환경에서 안전한 임무분리 정책을 구현할 수 있는 모델을 제안한다. 제안된 모델은 조직 구조를 보안 관리자의 관리 영역 기준으로 하는 OS-RBAC 모델을 기반으로 한다. 즉, OS-RBAC 모델에 임무분리 정책을 통합한 OS-SoDAM(Organizational Structure and Separation of Duty Administration Model)을 제안한다. 기존의 임무분리 정책들에 대해서 살펴보고, RBAC 모델에 적용했을 시 문제점과 본 연구의 동기에 대한 배경을 설명한다. 다음으로 OS-RBAC 모델과 위임 정책을 통합한 OS-DRAM 모델을 설명한다. 그런 후 OS-RBAC 모델 기반에 위임과 임무분리 정책을 통합한 OS-SoDAM 모델을 제안하고, 구현 시스템과 기존 RBAC 모델과 비교, 분석한 후 결론을 맺는다.

Ⅱ. 역할기반 접근제어 모델에서의 임무분리와 문제점

1. 역할기반 접근제어 모델에서의 임무분리 정책

역할 기반의 접근제어 정책에서 접근 결정은 개개인의 사용자들이 조직의 일부분으로서 가지는 역할에 근거한다. 인가권한은 역할 이름에 의해 그룹화되고, 자원의 이용은 그러한 권한에 연관된 역할이 부여된 개개인들에게로 제약된다[3]. 따라서 접근을 통제하기 위해 역할을 사용하는 것은 기업 특수의 보안 정책을 시행하고 개발하며 보안 관리를 능률화하기 위한 효율적인 도구가 된다[4]. 대규모의 기업 시스템에서는 역할의 수와 사용자의 수가 매우 많은데, 이러한 역할과 사용자 그리고 그들의 상호관계성을 관리하는 일은 매우 힘든 일이다. 이를 해결한 모델이 ARBAC97(Administrative RBAC)모델이다[5].

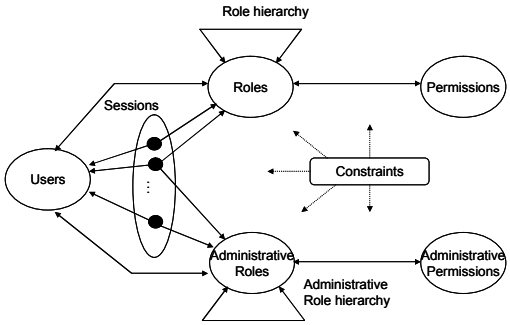

그림 1. ARBC97 모델
Fig. 1. ARBAC97 Model

역할기반 접근제어 모델에서는 사용자, 역할 및 인가권한 이렇게 세 가지 구성 요소를 갖추고, 사용자가 역할에 할당되고, 역할에는 인가권한들이 배정됨으로써 사용자가 자원에 접근할 수 있다. 여기서 제약사항(constraint)은 역할 기반의 접근제어 모델의 각 구성 요소들과 매핑들에 적용될 수 있는 것으로 임무분리, 한 역할에 지정될 수 있는 최대, 최소 사용자의 수(cadinality), 사용자가 특정 역할에 소속원이 되기 위한 선행 역할 그리고 사용자가 특정 작업을 수행함에 있어 필수 권한만을 부여하는 최소권한의 원리(least privilege principle) 등이 될 수 있다.

따라서 임무분리 제약조건을 해결하는 데 있어서 이해관계가 발생할 수 있는 부분을 세 분류로 구분할 수 있다[6, 7].

① 사용자 충돌(Conflicting users)
- 충돌 사용자들이 같은 역할에 할당되는 경우
- 충돌 사용자들이 충돌 역할에 각각 할당되는 경우

② 역할 충돌(conflicting roles)
- 충돌 역할들의 조상이 같은 경우
- 한 사용자가 충돌 역할에 할당되는 경우
- 충돌 역할에 동일한 인가권한이 할당되는 경우

③ 인가권한 충돌(Conflicting permissions)
- 충돌 인가권한들이 한 역할에 배정되는 경우

- 다른 역할에 각각의 인가권한이 할당되어 있더라도 그 각 역할에 동일한 사용자가 할당되는 경우

추가적으로, 일을 수행하는 사용자의 참여시간, 참여공간, 참여방법에 의해 제한될 수 있다[8, 9].

2. ARBAC 모델에서의 임무분리 정책 수행의 문제점

2.1 역할계층에 의한 임무분리 정책 위반

일반적으로 역할기반 접근제어 연구들에서는 임무분리를 역할 계층의 제약사항 때문에 구현의 관점에서 볼 때 공통적으로 다음과 같은 문제를 안고 있다.

2.1.1 사용자 충돌(Conflicting users)

충돌 사용자 u1과 u2는 같은 역할에 할당될 수 없는 임무분리 정책에 의해 역할 r1에 u1이 할당되면 u2는 할당되지 못한다. 그러나 u2가 r1의 상위 역할 x에 할당되면 역할계층의 계승 원칙에 의해 r1에 할당된 권한은 자동적으로 x에 계승됨으로써 암시적으로 u2는 r1의 권한을 얻게 된다.

또한, [그림 2]처럼 충돌 사용자 u1과 u2는 같은 충돌 역할에 각각 할당될 수 없다는 임무분리 정책에 의해 r1에 u1이 할당되면 u2는 r2에 할당되지 못한다. 그러나 u2가 r1과 r2의 상위 역할 x에 할당되면 역할계층의 특성에 의해 r1의 권한과 r2의 권한은 자동적으로 x에 계승됨으로써 암시적으로 u2는 r2의 권한을 얻게 된다.

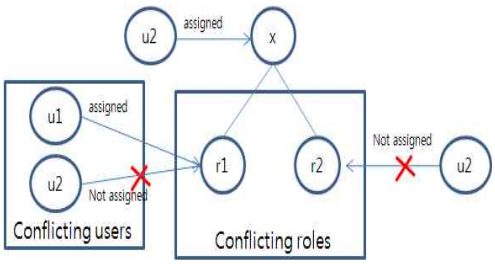

그림 2 충돌 사용자들이 충돌 역할에 할당되는 문제
Fig. 2. Problem of assigning conflicting users to conflicting roles

2.1.2 역할 충돌(Conflicting roles)

사용자 u1은 충돌 역할에 r1과 r2에 동시에 할당될 수 없는 임무분리 정책에 의해 r1에 u1이 할당되면 r2에는 u1이 할당되지 못한다. 그러나, u1이 r1과 r2의 상위 역할 x에 할당되면 역할계층의 특성에 의해 r2의 권한은 자동적으로 x에 계승됨으로써 암시적으로 u1는 r2의 권한을 얻게 된다[10].

또한, [그림 3]처럼 인가권한 p1은 충돌 역할에 r1과 r2에 동시에 할당될 수 없는 임무분리 정책에 의해 r1에 p1이 할당되면 r2에는 p1이 할당되지 못한다. 그러나 p1이 r2의 하위 역할 x에 할당되면 역할계층의 특성에 의해 r2의 권한은 자동적으로 x의 권한을 계승함으로써 암시적으로 r2는 p1 권한을 얻게 된다.

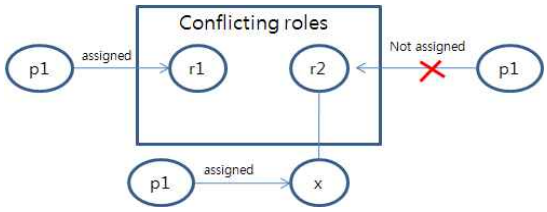

그림 3. 충돌 역할에 동일한 인가권한 할당되는 문제
Fig. 3. Problem of assigning a permission to conflicting roles

2.1.3 인가권한 충돌(Conflicting permissions)

충돌 인가권한 p1과 p2는 동시에 역할에 r1에 할당될 수 없는 임무분리 정책에 의해 r1에 p1이 할당되면 p2는 r1에 할당되지 못한다. 그러나 p2가 r1의 하위 역할 x에 할당되면 역할계층의 계승 원칙에 의해 r1의 권한은 자동적으로 x의 권한을 계승함으로써 암시적으로 r1는 충돌 인가권한 p1과 p2 모두를 얻게 된다[11].

또한, [그림 4]처럼 충돌 인가권한 p1과 p2는 각각 다른 역할 r1과 r2에 할당되었지만, 사용자 u1이 r1과 r2에 할당되거나 r1과 r2의 상위 역할이면서 조상이 되는 역할 x에 할당됨으로써 충돌 인가권한 p1과 p2를 동시에 할당 받게 된다.

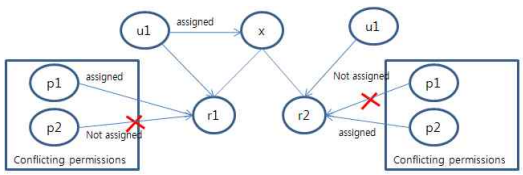

그림 4. 다른 역할에 다른 인가권한이 할당되는 문제
Fig. 4. Problem of assigning conflicting permissions to a same senior role

따라서 역할계층에 의한 임무분리 정책 위반 문제를 해결하기 위해서는 보안 관리자의 관리 영역 기준을 역할 계층이 아닌 다른 기준을 설정함으로써 역할 계층의 구성 요건을 완화시킬 수 있는 접근제어 모델이 필요하다.

2.2 임무분리 관리영역의 이탈 문제

2.1절에서 보았듯이 보안 관리자의 관리 영역 기준을 역할 계층으로 설정함으로써 충돌 역할들 간의 임무분리 정책은 그 역할들의 하위 역할 혹은 상위 역할에 의해 임무분리 정책이 위반될 수 있다.

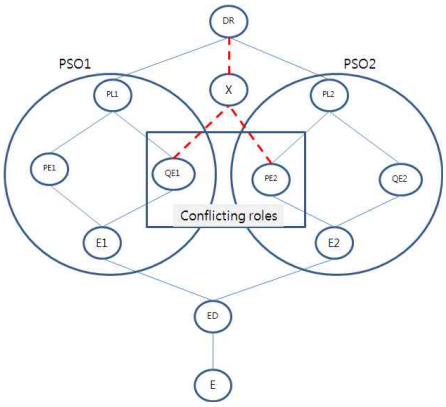

그림 5. 임무분리 관리 영역의 문제
Fig. 5. Problem of SOD authority range

[그림 5]처럼 보안관리자 PSO1과 PSO2는 자신의 관리 영역을 벗어나서 생성된 역할 x에 의해 충돌 역할 QE1과 PE2에 할당된 권한이 역할 x에 계승되어 임무분리 정책이 위반되는 상황을 제어하지 못하고 있다. 따라서 관리 영역 밖의 역할에 자신이 관리하는 권한들이 계승되는 것을 막는 관리적 기능이 포함된 접근제어 모델이 필요하다.

III. OS-RBAC 모델과 OS-DRAM 모델

일반적으로 기업환경에서 보안관리자의 관리 영역 기준은 역할계층이 아닌 조직 단위가 바람직하다. 여기서 말하는 조직 단위는 판매부서, 회계부서, 또는 프로젝트 팀 등을 의미한다. 그 조직 단위들의 계층 구조를 통해 조직 구조는 형성된다. 결국, 각 조직 단위는 임무를 수행하기 위해 직원과 인가권한을 포함하게 되고, 직원이 접근권한 모델의 주체가 되고, 인가권한이 접근권한 모델의 객체가 된다.

이와 같은 기본적인 개념을 바탕으로 OS-RBAC 모델이 제안되었다[13]. 첫 번째, 조직구조 관리 모델(Organizational Structure Administration Model)은 보안관리자에 의해 운용되는 조직의 생성/제거, 사용자/인가권한을 조직에 할당, 조직 계층 구성 등의 관리적 행위를 말한다. 조직구조 관리 모델은 관리적 행위들에 대한 관리 규칙으로 사용자를 조직에 할당하는 기능(User-Organization Assignment: UOA)과 인가권한을 조직에 할당하는 기능(Permission - Organization Assignment: POA), 그리고, 조직계층을 구성하는 기능(Organization-Organization Assignment: OOA)이 포함된다.

두 번째, 역할 관리 모델(Role Administration Model)은 보안관리자에 의해 운용되는 역할의 생성/제거, 사용자/인가권한을 역할에 할당, 역할계층 구성 등의 관리적 행위를 말한다. 역할 관리 모델 역시 이와 같은 관리적 행위들에 대한 관리 규칙으로 구성된다. 관리 규칙은 사용자를 역할에 할당하는 기능(User-Role Assignment: URA)과 인가권한을 역할에 할당하는 기능(Permission - Role Assignment: PRA), 그리고, 역할계층을 구성하는 기능(Role-Role Assignment: RRA)이 포함된다.

[그림 6]은 사용자 레벨의 위임 정책을 수행하도록 지원하는 OS-RBAC 모델을 확장한 OS-DRAM(Delegation Role Administration Model) 구조이다. 위임과 관련된 컴포넌트들은 위임역할의 생성 및 제거, UDRA(사용자-위임역할 할당), PDRA(인가권한-위임역할 할당)이다[14].

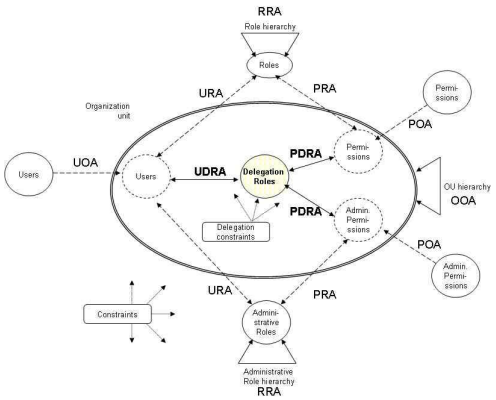

그림 6. OS-DRAM 모델
Fig. 6. OS-DRAM Model

OS-DRAM 모델은 OS-RBAC 모델에 위임 정책을 통합한 모델이지만, 임무분리 정책을 포함하고 있지 못하다. 본 논문에서는 OS-DRAM 모델에 2.1절에서 제시하고 있는 임무분리 문제를 해결하는 확장된 모델을 제안하고 구현을 제시하고 있다.

IV. OS-SoDAM 모델

1. 기본 구성요소

OS-SoDAM(Separation of Duty Administration Model) 모델은 OS-DRAM 모델에서 역할 관리 모델의 기능인 사용자-역할 할당, 인가권한-역할 할당, 역할-역할 할당뿐만 아니라 역할의 생성/제거, 인가권한의 생성/제거에 대한 관리 행위에 임무분리 정책을 추가한 모델이다. 기본적인 구성 요소는 다음과 같다.

- OT : 조직 단위 ot의 집합
- U : 사용자 u의 집합

- u.org_unit : 사용자가 소속된 조직 단위 속성

• R : 역할 r의 집합
  - r.type: 역할 유형에 대한 속성으로 정규역할 GR(general role) 혹은 관리역할 AR (administrative role)
  - r.group: 역할 그룹에 대한 속성으로 부서역할 DR (department role) 혹은 직책역할 JR (job role)
  - r.org_unit: 역할이 소속된 조직 단위 속성

• P : 인가권한 p의 집합
  - p.type: 인가권한 유형에 대한 속성으로 정규역할에 할당된 인가권한 GP (general perm) 혹은 관리역할에 할당된 인가권한 AP (admin. perm.)
  - p.org_unit: 인가권한이 소속된 조직 단위 속성

• DR: 위임역할 dr 의 집합
  - dr.org_unit: 조직 단위 속성
  - dr.creator: 위임역할 생성자 (사용자 ID, 정규역할)
  - dr.d_type: C(협업) 혹은 'B'(역할의 백업)

• R = RR ∪ DR: RR 은 OS-RBAC 모델에서의 정규 역할
• RR ∩ DR = ∅ 정규역할과 위임역할은 같은 역할군이 아님.
• 집합: U (사용자 데이터 집합), R (역할 데이터 집합), P (인가권한 데이터 집합), URA (사용자-역할 데이터 집합), PRA (인가권한-역할 데이터 집합)
• assigned-permissions(r): 임의의 역할 r(∈ R)에 할당된 인가권한
• active-permissions(r): 활성화 역할 r(∈ R)에 할당된 인가권한
• assigned-users(r): 임의의 역할 r(∈ R)에 할당된 사용자
• active-users(r): 활성화 역할 r(∈ R)에 할당된 사용자
• active-roles(u): 임의의 사용자 u(∈ U)에 의해 활성화된 역할
• seniors(r) = { x ∈ R | (x, r) ∈ RH}, 임의의 역할 r의 상위 역할을 출력하는 함수
• mutually-exclusive-permissions(p) = { x ∈ P | x is essential permissions require SoD of p}
• mutually-exclusive-roles(r) = { x ∈ R | x is essential roles require SoD of r}
• mutually-exclusive-users(u) = { x ∈ U | x is essential users required SoD of u}
• ∃dr ∈ DR, (dr.creator = (du, rr)) ∧ ((du, rr) ∈ URA) → assigned-permissions(dr) ∈ assigned-permissions(rr)

만약 위임자 du가 위임역할 dr을 생성한다면 자신의 역할 rr에 소속된 인가권한만을 위임할 수 있다.

2. 임무분리 제약 조건

역할은 사용자와 직접적인 연관을 맺는다. 해당 역할에 임의의 사용자를 할당할 때, 해당 사용자가 할당하고자 하는 역할과 충돌되는 다른 역할을 활성화하고 있는지 검사한다. 만약 가지고 있다면, 결과적으로 해당 사용자가 충돌되는 두 역할을 가지게끔 만들게 될 것이므로 이를 예방해야 한다.

2.1 사용자-역할 할당에서의 임무분리

[정의 4-1] (URA-SSOD 제약조건):

- 정적인 임무분리에 속한 충돌 사용자는 하나의 역할에 할당될 수 없다.

∀ u1, u2: U, r: R, u1 ≠ u2 :
{u1, u2} ∈ assigned-users(r) ∧
u1 ∉ mutually-exclusive-users(u2)

- 정적인 임무분리에 속한 충돌 사용자는 충돌 역할 각각에 할당될 수 없다.

∀ u1, u2: U, r1, r2: R, u1 ≠ u2, r1 ≠ r2 :
{u1, u2} ∈ assigned-users(r) ∧
u1 ∉ mutually-exclusive-users(u2) ∧
r1 ∉ mutually-exclusive-roles(r2) [

[정의 4-2] (URA-DSOD 제약조건):

- 동적인 임무분리에 속한 충돌 사용자는 동시에 하나의 역할에 할당될 수 없다.

∀ u1, u2: U, r: R, u1 ≠ u2 :
{u1, u2} ∈ assigned-users(r) ∧
{u1, u2} ∈ active-users(r) ∧
u1 ∉ mutually-exclusive-users(u2)

- 동적인 임무분리에 속한 충돌 사용자는 동시에 충돌 역할에 각각 할당될 수 없다.

∀ u1, u2: U, r1, r2: R, u1 ≠ u2, r1 ≠ r2 :
{u1, u2} ∈ assigned-users(r) ∧
{u1} ∈ active-users(r) ∧
{u2} ∉ active-users(r) ∧
u1 ∉ mutually-exclusive-users(u2) ∧
r1 ∉mutually-exclusive-roles(r2)

이와 같은 정적인 임무분리 및 동적인 임무분리에 속한 사용자-역할 할당에 대한 관리 행위는 각 보안관리자의 관리 영역을 벗어날 수 없다.

[보조정리 4-1] (UOA-SOD 제약조건): 임의의 조직단위에서의 URA-SSOD 및 URA-DSOD는 그 조직단위의 보안관리

자의 URA 관리 영역을 벗어나지 않는다.

{u1, u2} ∈assigned-users(r)는 다음과 같다.

(so.org_unit ≧ u1.org_unit ∧
so.org_unit ≧ r.org_unit ∧
u1.org_unit ≧ r.org_unit) ∧
(so.org_unit ≧ u2.org_unit ∧
so.org_unit ≧ r.org_unit ∧
u2.org_unit ≧ r.org_unit)

2.2 역할-역할 할당에서의 임무분리

[정의 4-3] (RRA-SSOD 제약조건):

- 정적인 임무분리에 속한 역할들에는 동일한 상위역할이 존재하지 않는다.

∀ r1, r2: R
r1 ∈ mutually-exclusive-roles(r2) ∧
(seniors(r1) ∩ seniors(r2) = ∅)

[정의 4-4] (RRA-DSOD 제약조건):

- 동적인 임무분리에 속한 역할들에는 동시에 동일한 상위역할이 존재하지 않는다.

∀ u: U, ∀ r1, r2: R, r1 ≠ r2
{r1, r2} ∈ active-roles(u) ∧
r1 ∉ mutually-exclusive-users(r2) ∧
(seniors(r1) ∩ seniors(r2) = ∅)

[보조정리 4-2] (ORRA-SOD 제약조건): 임의의 조직단위에서의 RRA-SSOD 및 RRA-DSOD는 그 조직단위의 보안관리자의 관리 영역을 벗어나지 않는다.

2.3 인가권한-역할 할당에서의 임무분리

[정의 4-5] (PRA-SSOD 제약조건):

- 정적인 임무분리에 속한 충돌 권한은 동일한 역할에 배정될 수 없다.

∀ p1, p2: P, r: R, p1 ≠ p2 :
{p1, p2} ∈ assigned-permissions(r) ∧
p1 ∉ mutually-exclusive-users(p2)

- 정적인 임무분리에 속한동일한 인가권한이 충돌 역할에 배정될 수 없다.

∀ p: P, r1, r2: R, r1 ≠ r2 :
{p} ∈ assigned-permissions(r1) ∧
{p} ∈ assigned-permissions(r2) ∧
r1 ∉ mutually-exclusive-roles(r2)

- 정적인 임무분리에 환경에서 다른 역할에 각각의 충돌 인가권한이 할당되어 있더라도 그 각 역할에 동일한 사용자가 할당될 수 없다.

∀ p1, p2: P, r1, r2: R, u: U, p1 ≠ p2, r1 ≠ r2 :
{p1} ∈ assigned-permissions(r1) ∧
{p2} ∈ assigned-permissions(r2) ∧
p1 ∈ mutually-exclusive-permissions(p2) ∧
{r1, r2} ∉ assigned-permissions(u)

[정의 4-6] (PRA-DSOD 제약조건):

- 동적인 임무분리에 속한 충돌 권한은 동시에 동일한 역할에 배정될 수 없다.

∀ p1, p2: P, r: R, p1 ≠ p2 :
{p1, p2} ∈ assigned-permissions(r) ∧
{p1, p2} ∈ active-permissions(r) ∧
p1 ∉ mutually-exclusive-users(p2)

- 동적인 임무분리에 속한 동일한 인가권한이 동시에 충돌 역할에 배정될 수 없다.

∀ p: P, r1, r2: R, r1 ≠ r2 :
{p} ∈ assigned-permissions(r1) ∧
{p} ∈ assigned-permissions(r2) ∧
{p} ∈ active-permissions(r1) ∧
{p} ∈ active-permissions(r2) ∧
r1 ∈ mutually-exclusive-roles(r2)

- 동적인 임무분리에 환경에서 다른 역할에 각각의 충돌 인가권한이 할당되어 있더라도 그 각 역할에 동시에 동일한 사용자가 할당될 수 없다.

∀ p1, p2: P, r1, r2: R, u: U, p1 ≠ p2, r1 ≠ r2 :
{p1} ∈ assigned-permissions(r1) ∧
{p2} ∈ assigned-permissions(r2) ∧
p1 ∈ mutually-exclusive-permissions(p2) ∧
{r1, r2} ∈ active-permissions(u)

[보조정리 4-3] (POA-SOD 제약조건): 임의의 조직단위에서의 PRA-SSOD 및 PRA-DSOD는 그 조직단위의 보안관리자의 PRA 관리 영역을 벗어나지 않는다.

3. 기존 임무분리 문제 해결

2.1절에서 제시한 역할계층에 의한 임무분리 정책 위반은 보안관리자의 관리 영역을 역할계층으로 했기 때문이다. 즉, 최상위 역할을 제외한 모든 역할은 부모역할과 자식역할 사이에 간선을 갖도록 하고 있기 때문에 역할계층은 트리 행태를 가질 수 없다.

반면, OS-DRAM 모델에서 역할의 'org_unit' 속성은 역할이 어떤 조직단위에 포함되어 있는지를 알려준다. 즉, 역할은 자신의 보안관리자를 갖고 있다. 따라서 보안관리자의 권한

범위는 역할계층에 종속하지 않는다. 결과적으로 OS-DRAM 모델은 [그림 7]처럼 다양한 유형의 역할계층을 제공한다.

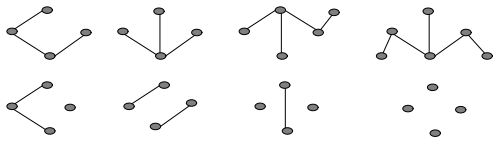

그림 7. OS-DRAM 모델에서 허용되는 역할계층 유형
Fig. 7. Types of role hierarchy of OS-DRAM model

3.1 역할계층에 의한 사용자 충돌에서의 임무분리 정책

충돌 사용자 u1과 u2는 같은 역할에 할당될 수 없는 임무분리 정책에 의해 r1에 u1이 할당되면 u2는 할당되지 못한다. 또한 [정의 4-1] URA-SSOD와 [정의 4-2] URA-DSOD에 의해 u2는 충돌 사용자 u1이 할당된 역할 r1의 모든 상위 역할에 할당될 수 없다.

[그림 8]처럼 충돌 사용자 u1, u2는 같은 충돌 역할에 각각 할당될 수 없다는 임무분리 정책에 의해 r1에 u1이 할당되면 u2는 r2에 할당되지 못한다. 또한 [정의 4-1]과 [정의 4-2]에 의해 u2가 r1과 r2의 상위 역할 x에 할당될 수 없다.

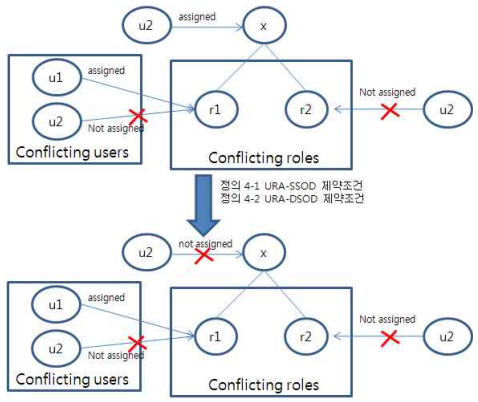

그림 8. 충돌 사용자들이 충돌 역할에 각각 할당되는 문제 해결
Fig. 8. Resolution of assigning conflicting users to conflicting roles

3.2 역할 충돌에서의 임무분리 정책

OS-SoDAM 모델에서 보안관리자의 권한 범위는 역할계층에 종속하지 않는다. [정의 4-3] RRS-SSOD 및 [정의 4-4] RRA-DSOD 제약조건에 의해 충돌 역할들 간에는 동일한 조상이 생성될 수 없다. 또한, 한 사용자는 충돌 역할에 할당될 수 없다.

또한, [그림 9]처럼 인가권한 p1은 충돌 역할에 r1과 r2에 동시에 할당될 수 없는 임무분리 정책에 의해 r1에 p1이 할당되면 r2에는 p1이 할당되지 못한다. 또한, [정의 4-5] PRA-SSOD 및 [정의 4-6] PRA-DSOD 제약조건에 의해 p1이 r2의 하위 역할 x에 할당될 수 없다.

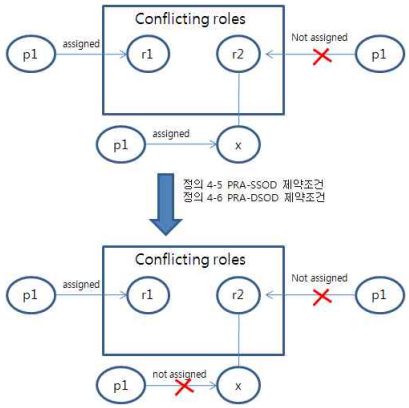

그림 9. 충돌 역할에 동일한 인가권한이 할당되는 문제 해결
Fig. 9. Resolution of assigning a permission to conflicting roles

3.3 인가권한 충돌에서의 임무분리 정책

충돌 인가권한 p1과 p2는 동시에 역할에 r1에 할당될 수 없는 임무분리 정책에 의해 r1에 p1이 할당되면 p2는 r1에 할당되지 못한다. 또한, [정의 4-5] PRA-SSOD 및 [정의 4-6] PRA-DSOD 제약조건에 의해 p2가 r1의 하위 역할 x에 할당될 수 없다.

또한, [그림 10]처럼 충돌 인가권한 p1과 p2는 각각 다른 역할 r1과 r2에 할당된 경우 [정의 4-3] RRA-SSOD 제약조건 및 [정의 4-4] RRA-DSOD 제약조건에 의해 역할 x가 생성될 수 없다.

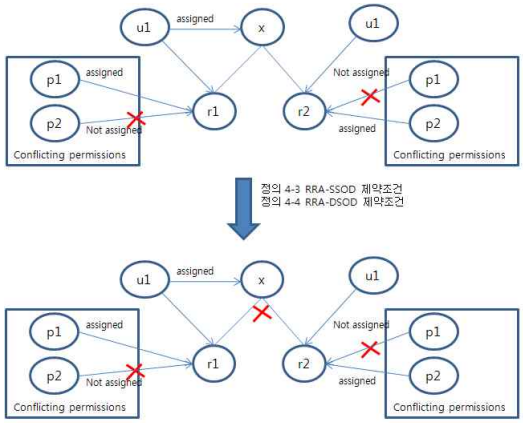

그림 10. 다른 역할에 각각의 인가권한이 할당되는 경우 동일 사용자 할당 문제 해결
Fig. 10. Resolution of assigning conflicting permissions to a same senior role

3.4 임무분리의 관리 영역 문제 해결

[보조정리 4-1 ~ 4-3]에 의해 일련의 사용자-역할 할당, 역할-역할 할당, 인가권한-역할 할당은 보안관리자의 관리영역을 벗어날 수 없다. 따라서 관리 영역 밖의 역할에 자신이 관리하는 권한들이 계승될 수 없다.

V. OS-SoDAM 구현 및 평가

1. 구현 내용

기존 OS-DRAM 엔진에 OS-SoDAM 모델의 제약조건을 통합한 OS-SoDAM 엔진을 이클립스 버전 3.1.1을 이용한 자바 언어를 통해 구현하였다.

OS-SoDAM 엔진은 두 부분으로 나뉜다:

① 모델 패키지: 충돌 사항을 등록, 수정 및 삭제 등의 관리 행위를 지원하는 사용자 인터페이스 클래스

② 제약조건 검사 패키지: 충돌 사용자, 충돌 역할 그리고 충돌 인가권한 등의 제약조건을 검사하면서 등록, 수정 및 삭제 등의 관리 행위를 지원하는 클래스

[그림 11]는 인가권한 'gen_p1'과 'gen_p2'이 충돌 인가권한이며 현재 '개발팀총괄역할'이라는 역할에 'gen_p1'이 할당되어 있는 상태에서 '개발팀총괄역할'에 'gen_p2'를 추가하려고 시도했을 때 충돌 역할 제약조건에 의해 에러가 발생되는 모습을 보여주고 있다. 이와 같은 구현을 통해 OS-SoDAM 모델이 현실세계에 잘 적용됨을 확인할 수 있다.

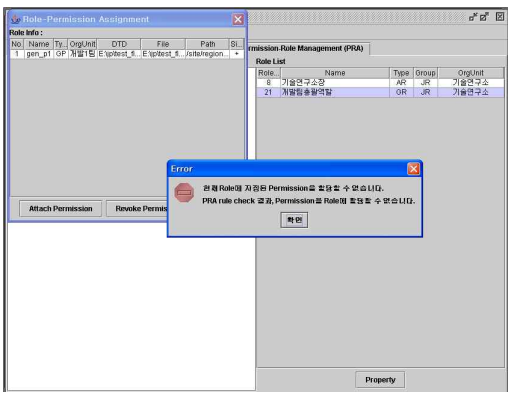

그림 11. 충돌 인가권한을 한 역할에 할당할 때의 에러 처리
Fig. 11. Error processing of assigning conflicting permissions to a same role

2. 평가

OS-SoDAM 모델은 기존의 다른 임무분리 통합 모델의 단점을 해결하고 있다. 기존 연구들은 RBAC96모델 혹은 ARBAC97 모델을 기반으로 임무분리 정책을 제안하고 있으며 OS-SoDAM 모델은 OS-RBAC 모델을 기반으로 위임 정책을 통합한 OS-DRAM 모델을 확장하여 임무분리 정책을 통합한 모델이다. 표 1은 기존 연구와 논문에서 제안한 연구와의 정성적인 비교를 보여주고 있다.

표 1. 다른 모델과 OS-SoDAM 모델의 비교
Table 1. Comparison OS-SoDAM with Other Models

	Streambeck[8]	Chen[9]	Moon[12]	OS-SoDAM
기본관리 모델	ARBAC97	ARBAC97	RBAC96	OS-RBAC
보안관리자의 관리영역	역할계층	역할계층	역할계층	조직 단위
URA 충돌 문제	No	No	Yes (제약조건 규칙 적용)	Yes (제약조건 규칙 적용)
PRA 충돌 문제	No	No	Yes (제약조건 규칙 적용)	Yes (제약조건 규칙 적용)
PRA 충돌 문제	No	No	Yes (제약조건 규칙 적용)	Yes (제약조건 규칙 적용)
불법적인 관리 행위 검사 기능	역할계층에 종속	역할계층에 종속	역할계층에 종속	조직 단위의 범위 선정으로 검사 가능
위임과 임무분리 통합	No	No	No	Yes
분산 환경 지원	Yes	Yes	No	Yes
구현 사항	No	No	No	Yes

VI. 결 론

본 연구에서는 다수의 보안 관리자에 의한 분산 접근제어 관리라는 필요성과 권한에 대한 임무분리라는 정책적 필요성을 만족시키기 위해 OS-SoDAM모델을 제안하였다. OS-RBAC 모델에 위임 정책과 임무분리 정책을 통합한 OS-SoDAM 모델은 위임과 임무분리를 통합 지원하는 장점을 갖고 있다. OS-RBAC 모델의 조직 구조 개념의 분산 접근제어 관리 개념을 그대로 계승하였기에 임무분리 정책 또한 관리영역과 역할계층을 분리하여 임무분리의 관리 영역 문제를 해결하였다.

일을 수행하는 사용자의 참여시간, 참여공간, 참여방법에 의해 임무분리를 제한하는 연구가 진행되었다[15, 16]. 향후 이와 같은 콘텐트 기반 임무분리 모델을 연구하고자 한다.

참고문헌

[1] M. J. Nash and K. R. Poland, “Some Conundrums Concerning Separation of Duty,” IEEE Symposium on Research in Security and Privacy, 7-9, pp. 201-209, 1990.

[2] V. D. Gligor, S. I. Gavrila and D. Ferraiolo, “On the Formal Definition of Separation-of-Duty Polices and their Composition,” IEEE Symposium on Security and Privacy, pp. 172-183, May, 1998.

[3] R. Sandhu, D. Ferraiolo, and D. Kuhn, “The NIST model for role-based access control: towards a unified standard”, in Proc. of Fifth ACM Workshop on Role-Based Access Control, pp. 47-63, 2000.

[4] 문형진, 서정석, “역할기반 접근제어시스템에 적용 가능한 민감한 개인정보 보호모델,” 한국컴퓨터정보학회 논문지, 제 13권, 제 5호, 103-110쪽, 2008년 9월

[5] R. Sandhu, V. Bhamidipati, and Q. Munawer, “The ARBAC97 model for role-based administration of roles,” ACM Trans. Inf. And Syst. Sec. 1, 2, pp. 105-135, 1999.

[6] S. Perelson, R. Botha and J. Eloff, “Separation of Duty Administration,” South African Computer Journal, Number 27, pp. 64 - 69, 2001.

[7] J. B. D. Joshi, E. Bertino, B. Shafiq, A. Ghafoor, “Dependencies and Separation of Duty Constraints in GTRBAC,” SACMAT'03, pp. 51-64, June 2003.

[8] M. Streambeck, “Conflict Checking of Separation of Duty Constraints in RBAC Implementation Experiences,” in Proc. of the Conference on Software Engineering (SE2004), pp. 224-229, Feb. 2004.

[9] H. Chen and N. Li, “Constraint Generation for Separation of Duty,” SACMAT'06, pp. 130-138, June 2006.

[10] T. Mossakowski, M. Drouineaud and K. Sohr, “A temporal-logic extension or role-based access control covering dynamic separation of duties,” 4th International Conference on Temporal Logic, pp. 83 - 90, July 2003.

[11] 오세종, “역할기반 접근제어 환경에서 접근권한 기반의 임무분리 모델,” 정보처리학회논문지 C, 제11-C권, 제 6호, 725-730쪽, 2004년 12월

[12] C. J. Moon, D. H. Park, S. J. Park, D. K. Baik, “Symmetric RBAC model that takes the separation of duty and role hierarchies into consideration,” Computers & Security, Vol 23, pp.126-136, 2004.

[13] Sejong Oh, Changwoo Byun, Seog Park "An Organizational Structure-Based Administration Model for Decentralized Access Control," Journal of Information Science and Engineering, Vol.22, No. 6, pp. 1465-1483, 2006.

[14] Changwoo Byun, Seog Park, Sejong Oh "OS-DRAM: A Delegation Administration Model in a Decentralized Enterprise Environment," The Seventh International Conference on Web-Age Information Management (WAIM 2006), Lecture Notes in Computer Science (LNCS)4016, pp. 593-604, June, 2006.

[15] 황유동, 박동규, “유비쿼터스 환경의 접근제어를 위한 확장된 GTRBAC 모델,” 한국 컴퓨터정보학회 논문지, 제 10권, 제 3호, 45-54쪽, 2005년 7월

[16] N. Li, and Q. Wang, “Beyond Separation of Duty: An Algebra for Specifying High-level Security Policies,” CCS'06, pp. 356-369, 2006.

저 자 소 개

변 창 우

2001 : 서강대학교 공학석사.

2007 : 서강대학교 공학박사.

2007 - 현재 : 인하공업전문대학 컴퓨터시스템과 교수

관심분야 : XML 접근제어, 접근제어 모델, 유비쿼터스 보안, 모바일 보안, 모바일 데이터 처리