분산 기업 환경을 위한 위임 관리 모델*

변창우O 박 석
서강대학교 컴퓨터학과
{changO, spark}@dblab.sogang.ac.kr

A Delegation Administration Model in a Decentralized Enterprise Environment

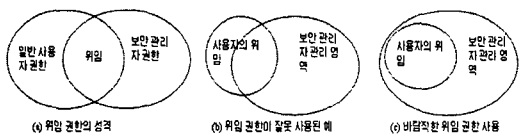
Chang-woo ByunO Seog Park
Department Computer Science , Sogang University

요 약

위임은 접근제어 분야에서 중요한 보안 정책 중의 하나이다. 사용자 레벨의 위임 관점에서 역할 기반 접근제어 모델을 기반으로 한 과거 위임 모델은 정보 노출과 위임 권한의 남용과 같은 보안 문제를 충분히 해결하지 못하고 있다. 본 논문에서는 관리적 역할기반 접근제어 모델과 위임 정책을 통합한 조직 구조 기반의 위임 역할 관리 모델(OS-DRAM)을 제시한다. OS-DRAM은 관리 행위의 범위 기준을 역할 계층과는 구별된 조직 구조로 정의하고 있어 사용자들에게 필요시 보안 관리자의 개입 없이 보안 관리자의 관리 범위 안에서 자유롭게 자신의 권한을 다른 사용자에게 위임할 수 있는 수단을 제공한다.

1. 서론

위임(delegation)이란 어떤 사정에 의하여 사용자가 다른 사용자에게 자신이 가진 권한의 일부 혹은 전부를 부여하여 자신의 업무를 대신 수행하게 하다가 필요시 부여한 권한을 다시 회수하는 행위를 말한다. 이와 같이 권한을 부여하고 회수하는 행위는 보안 관리자의 권한이기 때문에 사용자 레벨의 위임을 구현하기 위해서는 그림 1(a)와 같이 보안 관리자의 권한 영역을 일부 침범한다고 볼 수 있다. 따라서, 위임 권한이 잘못 사용될 경우 그림 1(b)와 같이 보안 관리자의 관리 영역을 벗어날 수 있다. 현실적으로는 그림 1(c)와 같이 사용자가 일정 범위 내에서 보안 관리자의 개입 없이 위임 권한을 행사할 수 있으면서도 전체적으로는 보안 관리자의 관리 영역을 벗어나지 않도록 하는 위임 모델이 필요하다.



(a) 위임 권한의 성격 (b) 위임 권한이 잘못 사용된 예 (c) 바람직한 위임 권한 사용

그림 1. 위임 권한의 성격과 사용

과거 역할기반 접근제어 모델[1]을 기반으로 한 관리적 역할기반 접근제어 모델[2] 상의 위임 모델들[4,5,6]은 보안 관리자의 관리 영역 기준을 역할 계층으로 하였기 때문에 역할 계층에 의해 발생되는 문제점을 그대로 앉고 있다.

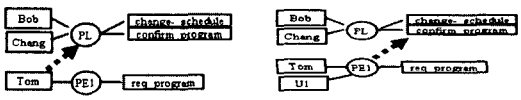
본 논문은 다수의 보안 관리자가 필요한 분산 접근제어 환경에서 안전한 위임을 구현할 수 있는 모델을 제안한다. 제안된 모델은 조직 구조를 보안 관리자의 관리 영역 기준으로 하는 OS-RBAC 모델[3]을 기반으로 위임 정책을 통합한 OS-DRAM(Organizational Structure and Delegation Role Administration Model)을 제안한다. 2장에서는 기존의 위임 모델의 문제점과 본 연구의 동기에 대한 배경을 설명한다. 3장에서는 OS-RBAC 모델과 위임 정책을 통합한 OS-DRAM을 제안한다. 4장에서 결론을 내린다.

2. 기존 연구 및 동기

역할기반 위임 모델(RBDM)[4]과 역할 위임 모델 2000 (RDM2000) [5]에서 제안하고 있는 위임 방법은 URA 방법과 부분 위임을 위한 PRA 방법이다. 그림 2(a) 는 역할 PL 에 할당된 Bob 이 자신의 역할을 Tom 에게 위임하는 URA 방법으로써 부분 위임을 할 수 없다는 문제점과 위임을 받은 Tom 이 PL 이하의 하부 역할까지 계승할 수 있는 문제점이 있다. 그림 2(b)는 부분 위임을 하기 위해 Bob 이 PL 역할에 할당된 'confirm_program' 인가권한만을 Tom 에게 위임하도록 하는 PRA 방법으로써 PE1 에 할당된 또 다른 사용자 U1 역시 접근할 수 있는 문제점이 발생한다. 결과적으로 URA 방법이나 PRA 방법은 '최소 권한의

* 본 연구는 정보통신부 대학 IT 연구센터 육성, 지원사업의 연구결과로 수행되었습니다.

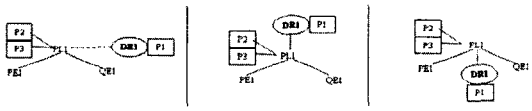
원칙(least privilege principle)' 을 위반하는 방법이다.



(a) URA 방법 (b) PRA 방법

그림 2. RBAC 상에서의 간단한 위임 방법

인가권한 기반의 위임 모델(PBDM)[6]에서는 ARBAC97 모델[2]을 기반으로 위임역할을 생성하고 위임받는 사용자와 위임하고자 하는 대상을 위임역할에 할당하는 방법을 제안하고 있다. ARBAC97 모델을 기반으로 하고 있기 때문에서 생성된 위임역할은 반드시 역할 계층에 소속되어야 한다. 그림 3(b)는 정규역할의 부모로 위임역할을 위치시키는 방법으로 URA 방법의 문제점을 그대로 안고 있다. 그림 3(c)는 정규역할의 자식으로 위임역할을 위치시키는 방법으로 PRA 방법의 문제점을 갖게 된다. 그림 3(a)는 이런 문제점들을 해결하기 위해 위임역할과 역할계층을 분리시키는 방법인데 ARBAC97 모델의 제약사항에 위배되는 방법이다.



(a)정규역할과 독립 (b)정규역할의 부모 (c)정규역할의 자식

그림 3. 정규역할과 위임역할의 위치

한편, ARBAC97 모델은 관리 데이터를 기반으로 관리 행위를 제한하고 있는데, 잘못된 관리 데이터를 판단하는 기준이 없는 문제점을 앉고 있다. 이 문제는 위임 모델에서 위임 행위를 제한하는 *can-delegate* 제약사항에 영향을 준다.

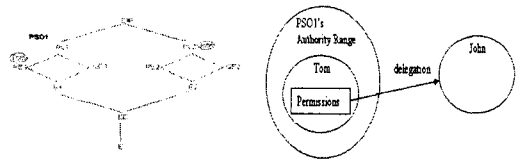


그림 4. 관리 데이터 및 역할계층에 의한 위임의 일탈

can-delegate (PE1, ED, req_ program, 1) 규칙을 갖고 그림 4처럼 Tom이 'req_program' 인가권한을 역할 PE2에 할당된 John에게 위임한다고 가정하자. PE2는 역할 ED의 상위 역할이므로 John은 ED에 속한다. 이 경우 Tom은 보안 관리역할 PSO1의 관리 하에 있지만 John은 PSO1의 관리 영역 밖에 있기 때문에 Tom의 권한이 John에 의해 잘못 사용될 수 있다.

위임 모델이 현실에 적용될 수 있기 위해서는 위에 언급한 문제점들이 해결되어야 한다. 다음 장에서는 이 문제를 해결할 수 있는 위임 역할 관리 모델을 제안한다.

3. 조직 구조 기반의 위임역할 관리 모델: OS-DRAM

그림 5는 사용자 레벨의 위임 정책을 수행하도록 지원하는 OS-RBAC 모델을 확장한 OS-DRAM 구조이다. 위임과 관련된 컴포넌트들은 위임역할의 생성 및 제거, UDRA(사용자-위임역할 할당), PDRA(인가권한-위임역할 할당)이다. 자세히 설명하기 전에 OS-DRAM의 용어를 간단히 설명한다.

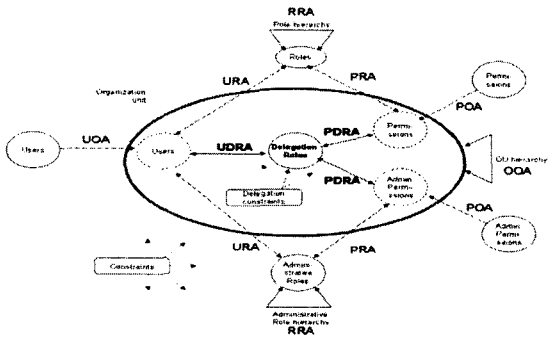


그림 5. OS-DRAM

- *U*: 사용자 *u* 의 집합, *u.org_unit*: 조직 단위 속성
- *P*: 인가권한 *p* 의 집합, *p.org_unit*: 조직 단위 속성
- *R*: 역할 *r* 의 집합, *r.org_unit*: 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 = Ø: 정규역할과 위임역할은 같은 역할군이 아님.
- *Permission(r)*: R → 2^P, a function mapping a role (regular role or delegation role) to a set of permissions
- 집합: U (사용자 데이터 집합), R (역할 데이터 집합), P (인가권한 데이터 집합), URA (사용자-역할 데이터 집합), PRA (인가권한-역할 데이터 집합)
- ∃*dr* ∈ DR, (*dr.creator* = (*du, rr*)) ∧ ((*du, rr*) ∈ URA) → Permissions(*dr*) ∈ Permission(*rr*)
  - 만약 위임자 *du* 가 위임역할 *dr* 을 생성한다면 자신의 역할 *rr*에 소속된 인가권한만을 위임할 수 있다.

3.1 위임역할 생성 및 제거 규칙

[D-Rule 1] 정규 역할 *r* 에 소속된 위임자 *DU* 가 위임역할 *DR* 를 생성할 수 있는 조건:

- (*DU.org_unit* ≥ *DR.org_unit*) ∧ (*DR.creator* = (*DU*, *r*))

[D-Rule 2] 위임자 *DU*가 위임역할 *DR*을 제거할 수 조건:

- (*DR.creator* = (*DU*, *r*)) ∧ (*DU.org_unit* ≥ *DR.org_unit*)

3.2 PDRA (Permission-Delegation Role Assignment)

[D-Rule 3] 위임자 *DU*가 위임역할 *DR*에 인가권한 *p*를 할당할 수 있는 조건:

- (*DR.creator* = (*DU*, *r*)) ∧ ((*DU*, *r*) ∈ URA) ∧ ((*p*, *r*) ∈ PRA) ∧ (*DU.org_unit* ≥ *DR.org_unit*) ∧ (*DR.org_unit* ≥ *p.org_unit*)

[D-Rule 4] 위임자 *DU*가 위임역할 *DR*에 인가권한 *p*를 제거할 수 있는 조건:

- (*DR.creator* = (*DU*, *r*)) ∧ (*DU.org_unit* ≥ *DR.org_unit*)

3.3 UDRA (User-Delegation Role Assignment)

[D-Rule 5] 위임자 *DU* 가 위임역할 *DR* 에 위임받는 사용자 *DDU*를 할당하기 위한 조건:

- (*DR.creator* = (*DU*, *r*)) ∧ (*DU.org_unit* ≥ *DR.org_unit*) ∧ (*DDU.org_unit* ≥ *DR.org_unit*) ∧ (*DU.org_unit* ≥ *p.org_unit*) ∧ (*DDU.org_unit* ≥ *p.org_unit*) ∧ (*DU.org_unit* ≥ *DDU.org_unit*)

[D-Rule 6] 위임자 *DU* 가 위임받은 사용자 *DDU* 로부터 위임역할 *DR*을 회수할 수 있는 조건:

- (*DR.creator* = (*DU*, *r*)) ∧ (*DU.org_unit* ≥ *DDU.org_unit*) ∧ (*DU.org_unit* ≥ *DR.org_unit*)

3.4 위임 권한부여 생성 규칙

위임 권한부여 시 *can-delegate* 제약사항의 목적은 위임 활동의 영역을 제한하는 것이다. OS-DRAM 의 *can-delegate* 제약사항은 다음과 같다.

▪ ***can-delegate* 제약사항**: (*rr*, *pc*, *s*, *n*) ∈ *can-delegate* 정규역할 *rr* (혹은 *rr*의 *상위역할*)에 할당된 사용자는 인가권한 (위임 영역) *s*를 *예비조건 pc*를 *만족하는 사용자에게 위임할 수 있다*. 추가로, 재위임은 최대 *n*번 *가능하다*. *이와 같은* ***can-delegate* 제약사항의 생성은 같은 보안 관리자의 관리권한 영역에서만 가능하다.**

[D-Rule 7] (*can-delegate* 제약사항 생성 조건)

(*SO.org_unit* ≥ *rr.org_unit*) ∧ (*SO.org_unit* ≥ *max*{*pc.org_unit*}) ∧ ((∃ *u1*, *u2* ∈ U, (*u1*, *rr*), (*u2*, *pc*) ∈ URA → (*u1.org_unit* ≥ max{*pc.org_unit*})) ∧ (*u1.org_unit* ≥ *u2.org_unit*)) ∧ (∃ {*s*} ⊆ P, *rr.org_unit* ≥ max{*s.org_unit*}) ∧ ({*s*} ⊆ *Permission* (*rr*))

추가적인 규칙은 위임 유형[8]에 관한 것이다. '역할의 백업' 을 위한 위임이라면 별개의 조직에 속한 사용자에게 권한을 위임하는 것은 바람직하지 않다. 그러나 다른 조직에 있는 사람과의 협업을 위한 것이라면 위임은 허용될 수 있다.

[D-Rule 8] 위임 유형에 따른 추가 조건:

- (*DR.d_type* = 'B') ∧ (*DR.creator* = (*DU*, *r*))
- (*DR.d_type* = 'C') ∧ (*DR.creator* = (*SO*(*security officer*), admin. R) ∧([D-Rule 7] is disregarded)

4. 결론 및 추후 연구

본 논문에서는 다수의 보안 관리자에 의한 분산 접근제어 관리라는 필요성과 권한의 위임이라는 정책적 필요성을 만족시키기 위해 OS-RBAC 모델에 통합된 OS-DRAM 을 제안하였다. OS-DRAM 이 효율적으로 분산 접근제어 관리를 지원할 수 있는 원인은 조직 구조라는 개념을 도입하여 위임 활동의 관리영역과 역할계층의 분리에 있다.

다단계 위임이 미치는 영향과 안전한 다단계 위임을 위한 추가적인 연구가 필요하다. 또한, 새로운 위임 관리 모델에서의 임무 분리(separation of duty)에 대한 연구도 필요하다.

5. 참고문헌

1. R. Sandhu, D. Ferraiolo, and D. Kuhn, "The NIST model for role-based access control: towards a unified standard" , *Proc. of Fifth ACM Workshop on Role-Based Access Control*, pp. 47-63.
2. 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.
3. S. Oh, C. Byun, and S. Park, "An Organizational Structure-Based Administration Model for Decentralized Access Control" , Journal of Information Science and Engineering, 2005(submitted).
4. E. Barka and R. Sandhu, "A Role-Based Delegation Model and Some Extensions" , *Proc. Of 23rd National Information Systems Security Conference (NISSC) 2000.*
5. Longhua Zhang, Gail-Joon Ahn, Bei-Tseng Chu, "A Rule-Based Framework for Role-Based Delegation and Revocation" , ACM Transactions on Information and System Security, Vol.6, No.3, pp.404-441, Aug. 2004.
6. Xinwen Zhang, Sejong Oh and Ravi Sandhu, "PBDM: A Flexible Delegation Model in RBAC" , Proc. 8th ACM Symposium on Access Control Models and Technologies (SACMAT), pp.149-157, 2003.