T-RBAC에 기초한 세션기반의 동적 의무분리

배혜진[0] 박석
서강대학교 컴퓨터학과 데이터베이스 연구실
(hjbae[0], spark)@dblab.sogang.ac.kr

Session-Based Dynamic Separation of Duty Using T-RBAC

Hye-Jin Bae[0] Seog Park
Database Research Lab., Dept. of Computer Science, Sogang University

요 약

의무분리 정책의 목적은 정보의 무결성을 필요로 하는 연산들을 여러 역할이나 사용자에게 분산시킴으로써 조직 내에서 관리하는 정보의 무결성 침해 가능성을 최소화하는 것이며, 이는 기업 환경에서 중요한 보안 요구사항이다. 역할기반 접근제어는 응용에 따라 보호 객체들에 대한 접근을 역할들로 분류하여 단순한 권한 관리를 제공하며, 의무분리 정책을 시행하기에 적합하여 기존의 강제적 접근제어나 임의적 접근제어에 대한 대안으로 의무분리와 관련하여 다양한 기법들이 제시되었다. 그러나 역할 수준의 의무분리는 역할에 할당된 과업들을 상호 배타적인 작업의 수행에 관련되지 않은 과업도 모두 배제시키게 되어 과업 실행의 유연성이 떨어지게 되므로 상호 배타적인 작업을 수행하는 과업들에 할당된 최소의 권한을 배제시키는 것이 합리적이다. 본 논문은 기업 환경에 적합한 과업-역할기반 접근제어 모델을 기초로 하여 과업의 특성에 따라 분류된 유형별로 과업 수준의 동적 의무분리를 적용하는 기법을 제시한다. 특히 실제 사용자가 병렬적으로 수행하는 워크플로우와 다중 세션 환경에서 상호 배타적인 과업들과 과업 인스턴스들에도 적용이 가능한 세션기반의 동적 의무분리 기법을 제시한다. 이때 기존의 동일 사용자에 의한 동적 의무분리 적용을 공모가 가능한 사용자들에 의해 생성된 다중 세션들간의 동적 의무분리를 제시함으로써 의무분리의 목적을 만족시킨다.

1. 서 론

최근 인터넷의 발달과 더불어 데이터의 양은 급속도로 증가하여 이를 효율적으로 관리하는 것 뿐만 아니라 이에 대한 적절한 보안 정책을 수립하는 것에 대한 중요성이 강조되고 있다. 역할기반 접근제어(Role-Based Access Control: RBAC)는 사용자 수가 많고 실제 정보에 대한 접근 통제가 제한된 수의 관리자에 의해 이루어지는 기업 환경에 적합한 방법이다. 사용자에게 직접 객체에 대한 접근 권한이 부여되는 것이 아니라 사용자에게는 실제 기업의 조직체계와 유사한 역할이 부여된다. 이 역할에 연산을 수행할 수 있는 권한들의 집합을 지정함으로써 권한 변경과 같은 관리를 단순화하고 여러 보안 정책들을 적용할 수 있는 정책 중립적인 특성을 가지고 있다[1].

RBAC의 여러 제약사항 중에 상업적 응용들에서 중요한 보안 요구사항인 데이터의 무결성을 필요로 하는 연산들을 여러 역할이나 사용자에게 분산시켜 공모, 사기와 같은 부정행위를 막는 의무분리에 관한 연구들이 진행되어 왔다. 이러한 연구들은 의무분리 정책을 적용하는 시점, 적용 대상, 정형 기술을 중심으로 진행되어 왔다[2,3,4,5,6].

최근에 제안된 RBAC에 대한 NIST 표준[7]은 정적 의무분리는 사용자의 전체 권한에서 제약사항을 정의하고 두는데 반해 동적 의무분리는 사용자의 세션 내부와 세션간에서 활성화 될 수 있는 역할들에 제약사항을 둠으로써 사용자의 전체 권한에서 권한의 사용을 제한한다. 이는 수행되고 있는 역할에 따라 사용자는 매번 차별화된 수준의 권한을 가지게 되어 최소 권한의 원리에 부합한다고 서술하고 있다. 따라서 동적 의무분리를 위해 세션기반의 기법의 필요성이 계속해서 언급되었다. 그러나 동일 사용자에 의해 생성되는 세션에서 활성화 되는 역할 집합이 동일하지 않은 경우를 상호 배타적인 세션으로 정의하고 동일 사용자에 의해 상호 배타적인 세션이 생성될 수 없도록 하는 방법[8]이나 단일 역할에 대해서 실행시 한 세션내에서 상호 배타적인 관계에 있는 프로시져들이 활성화 되지 않도록 동적으로 사용자의 주체등급이 결정되는 방법[9]은 다양한 응용을 수행하는 환경에서 병행성의 제약을 준다.

즉 역할 수준의 의무분리는 역할에 할당된 과업(task)들을 상호 배타적인 작업을 수행하는데 관련되지 않은 과업을 모두 배제시키게 되어 과업 실행의 유연성이 떨어지게 되므로 상호 배타적인 작업을 수행하는 과업들에 할당된 최소의 권한을 배제시키는 과업 수준의 의무분리를 적용하고 상호 배타적인 과업을 포함하는 세션 내부와 세션들간의 의무분리를 적용하는 것이 합리적이다.

2. T-RBAC에 기초한 세션기반의 동적 의무분리

2.1 과업의 재분류

본 논문은 RBAC의 개선된 모델인 T-RBAC모델[10]에 기반을 두고 있다. <그림 1>에서와 같이 권한을 역할에 직접 지정하는 것이 아니라, 역할보다 세분화된, 기업 환경에서 작업의 기본 단위인 과업을 두어 권한을 이러한 과업에 지정하며 과업이 역할에 지정된다. 또한 과업을 기업에서 발생할 수 있는 과업들의 특성에 기초하여 크게 네가지로 분류한 것이나 과업 분류는 기업의 보안 요구사항에 따라 달라질 수 있다는 유연성을 강조하고 있다. 본 논문에서 제안하는 과업 수준의 의무분리를 적용하기 위해 워크플로우에 속하는 과업을 class W로, 그렇지 않은 과업을 class NW로 재분류한다. 이는 워크플로우에서의 과업의 경우 과업의 인스턴스들간의 동적 의무분리도 중요한 고려사항이고 지금까지 이를 위한 동적 의무분리에 대한 연구가 거의 이루어지지 않아왔다. 따라서 과업의 재분류에 의한 수정된 T-RBAC모델을 기반으로 모든 논의가 이루어진다.

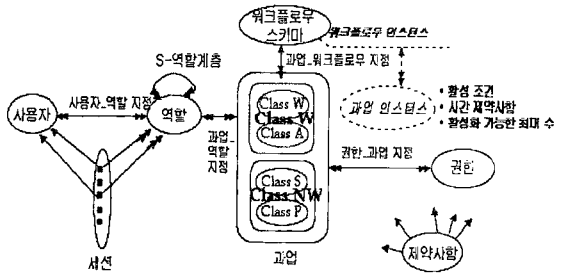

그림 1 T-RBAC에서 과업 재분류

2.2 제안한 세션기반의 동적 의무분리 기법

2.2.1 세션내부에서의 동적 의무분리

본 논문에서는 세션내부와 세션간의 동적 의무분리를 적용하기 위해 동일 사용자에 의해서만 적용하는 것보다 공모가 가능한 사용자의 집합으로 확장하여 고려하는데 이는 무결성 보장에 적합하다.우선 단일 세션내에서 상호 배타적인 과업이 동일한 역할에 할당된 경우와 상호 배타적인 과업이 다른 역할에 할당된 경우인데 역할 수준에서 상호 배타적인 역할일 경우가 대부분 이에 속하는 것으로 제안한 기법에서는 역할 전체가 아닌 상호 배타적인 과업을 동시에 활성화 되지 못하는 과업 스키마 수준의 동적 의무분리(TS-DSOD)로 표현된다. 또한 TS-DSOD에서의 과업이 class W에 속하는 과업인 경우 워크플로우 인스턴스에서도 과업 인스턴스 수준의 동적 의무분리(TI-DSOD)가 적용되는데 이를 형식적으로 기술하면 다음과 같다.

CU: Conflict User Set,
$CU_i \in CU$, $U_i \in CUi$, $R_i, Rj \in R$,
$Ri,Rj \in assigned(Ui)$, $Si \in created\_session(Ui)$,
$Ri,Rj \in session\_roles(Si)$, $T_i, T_j \in T$, $T_i \neq T_j$,
• [TS-DSOD] 과업 스키마 수준의 동적 의무분리
$(T_i \in assigned(R_i) \vee T_i \in assigned(Rj)) \wedge (T_j \in assigned(R_i) \vee T_j \in assigned(Rj)) \wedge activate(T_i) \wedge activate(T_j) \Rightarrow T_j \notin mutually\text{-}exclusive\text{-}task(T_i)$
• [TI-DSOD] 과업 인스턴스 수준의 동적 의무분리
$T_i, T_j \in class\ W$, $T_i, T_j \in same\text{-}instance\text{-}of\text{-}W$,
$(T_i \in assigned(R_i) \vee T_i \in assigned(Rj)) \wedge (T_j \in assigned(R_i) \vee T_j \in assigned(Rj)) \wedge activate(T_i) \wedge activate(T_j) \Rightarrow T_j \notin mutually\text{-}exclusive\text{-}task(T_i)$

2.2.2 세션간의 동적 의무분리

동일 사용자 또는 공모 가능한 사용자들에 의해 생성된 세션간에 상호 배타적인 과업이 동시에 활성화될 수 없는 세션간의 과업 스키마 수준의 동적 의무분리(MTS-DSOD)와 MTS-DSOD에서 과업이 class W에 속하는 상호 배타적인 과업의 인스턴스에도 세션내에 선행하는 상호 배타적인 과업이 완료된 경우에 세션간의 과업 인스턴스 수준의 동적 의무분리(MTI-DSOD)를 적용한다. 이를 형식적으로 기술하면 다음과 같다.

$CU_i \in CU$, $U_i \in CUi$, $R_i, Rj \in R$,
$Ri,Rj \in assigned(CUi)$, $Si,Sj \in created\_session(CUi)$, $Si \neq Sj$,
$Ri \in session\_roles(Si)$, $Rj \in session\_roles(Sj)$,
$T_i, T_j \in T$, $T_i \neq T_j$,
• [MTS-DSOD] 세션간의 과업 스키마 수준의 동적 의무분리
$(T_i \in assigned(R_i) \wedge T_j \in assigned(Rj) \wedge activate(T_i) \wedge activate(T_j) \Rightarrow T_j \notin mutually\text{-}exclusive\text{-}task(T_i)$
• [MTI-DSOD] 세션간의 과업 인스턴스 수준의 동적 의무분리
$T_i, T_j \in class\ W$, $T_i, T_j \in same\text{-}instance\text{-}of\text{-}W$,
$T_i \in assigned(R_i) \wedge T_j \in assigned(Rj) \wedge activate(T_i) \wedge activate(T_j) \Rightarrow T_j \notin mutually\text{-}exclusive\text{-}task(T_i)$

2.3 제안한 기법의 적용방법

■ 시스템에서 제안한 기법을 적용하기 위한 시나리오

1.사용자는 접근제어를 위해 세션을 생성한다.
2.미리 작성된 공모 가능한 사용자 집합에 속한 사용자들이 생성한 세션의 풀인 세션 도메인을 만든다.
3.<그림 2>의 세션 매니저는 실시간 제약사항이 런 타임 엔진에 의해 일관성 체킹후에 미리 Task design tool에서 추출된 정보를 기반으로 구성된 접근제어 카탈로그 정보와 각 사용자의 history정보 체킹, 세션 도메인 정보를 바탕으로 동적 의무분리를 확인한다.
4.만약 의무분리가 적용되어야 하는 과업은 세션 매니저에 의해 상호 배타적인 과업중 하나를 활성화 시키고 런타임 엔진에 의한 접근제어를 시도하도록 하고 의무분리 고려사항이 없는 과업은 런타임 엔진에 의해 접근제어가 시도된다.

```
Algorithm: Session Manager
Input:   1) Information of task design
         2) Runtime specification, constraints and code
         3) Information of SessionDomain ,history and user
Output: deactivated if dynamic separation of duty is not satisfied;
        otherwise.
1.Let n be the number of tasks in specific SD
2. Tj ≠ Tk and > represents execution order in workflow i.e.
        if Tj > Tk , then Tj is a prior task of Tk
Procedure SBDSOD (Tj,Tk)
If consistency and num_session(SD) ≠ 0:
For each SD in SessionDomain:
If Tj,Tk ∈ same session:
  apply task schema level DSOD[TS-DSOD];
  If Tj,Tk ∈ class W:
  apply task instance level DSOD[TI-DSOD];
Endif
If Tj,Tk ∉ same session:
  For Si ∈ SD:
    while i ≤ n
      If Tk ∈ mutually-exclusive-task (Tj):
        If activate(Tj) and activate (Tk):
          deactivate(Tj) or deactivate(Tk );
        If Tj,Tk ∈ class W and Tj,Tk ∈ same instance of W
            and Tj > Tk and activate (Tj):
          For Sl≠Si and l≠ i ≤ n :
            If completed (Tj) and activate (Tk):
              deactivate(Tj) or deactivate((Tk );
Endif
```

그림 2 Session Manager 알고리즘

3. 해결방안 및 평가

표 1 동적 의무분리 시 요구되는 사항과 이의 해결방안

요구 사항	해결 방안
과업의 재분류	T-RBAC모델에 기초하여 동적 의무분리 적용에 적합하도록 과업을 재분류 한다.
세션간의 동적 의무분리	다중 세션이 생성되었을 경우 과업분류에 기초하여 세션 내부와 세션을 교차하여 과업들간의 동적 의무분리를 적용한다.
인스턴스 수준의 동적 의무분리	워크플로우에서 class W에 속하는 과업에 의해 생성된 인스턴스들간의 동적 의무분리를 적용하고 다중 세션상에서도 이를 적용한다.
공모가 가능한 사용자 집합	동일 사용자에 의해 생성된 세션간의 동적 의무분리 뿐만 아니라 공모가 가능한 사용자들의 집합에 의해 생성된 세션간의 동적 의무분리를 적용한다.
구현 용이	T-RBAC 모델을 바탕으로 과업을 분류하여 활성화된 역할에 지정된 과업들의 특성을 쉽게 확인할 수 있으며 사용자 확인과 세션간의 구분은 구현상 용이하다.

기존의 RBAC과 워크플로우에 의무분리의 제약사항을 두기 위한 많은 기법들이 연구되었으나 역할계층상의 상위역할이 하위역할의 모든 권한을 상속 받음으로 인해 생기는 문제를 해결하고 기업 환경에서 만연하는 워크플로우를 고려한 접근제어를 가능하게 하는 T-RBAC모델을 바탕으로 과업 수준의 의무분리를 제시하였다. 또한 <표 1>에서 보는 바와 같이 동적 의무분리를 적용하는데 있어서 간과되어 오던 세션간의 적용방법과 공모 가능한 사용자 집합으로의 확장을 통해 의무분리의 목적을 향상시킬 수 있었다.

4. 결론

최근 기업의 보안 환경에 적합한 RBAC은 의무분리와 같은 의미적 제약사항의 적용에 대한 연구가 활발히 진행되어 왔다. 본 논문은 다양한 응용들을 수행하는 환경에서 동일 사용자 또는 공모 가능한 사용자들에 의해 생성된 세션내부와 세션간에서 역할 수준의 의무분리보다 효율적으로 의무분리를 시행할 수 있는 과업수준의 의무분리 기법을 제시하였다. 실제 워크플로우 시스템의 적용과 다른 세션에 속한 과업들의 인스턴스 사이의 의존,관계 등을 기술하는 방법,지원 기법과 적절한 history 관리방안에 대한 추가적인 연구가 필요하다.

5. 참고문헌

[1] Ravi Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E.Youman, "Role-Based Access Control Models", *IEEE Computer*, Vol.29, No. 2, pp. 38~47,1996.

[2] Simon R.T and Zurko M.E " Separation of Duty in Role-Based Environments" , *Proceedings of Computer Foundations Workshop X*, 1997.

[3] Richard Kuhn, "Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems', *Second ACM Workshop on Role-Based Access Control*, 1997.

[4] Gail-Joon Ahn and Ravi Sandhu, " Role-Based Authorization Constraints Specification", *ACM Transactions on Information and Systems Security*, Vol.3, No.4, 2000.

[5] Ravi Sandhu, " Transaction Control Expressions for Separation of Duties" *Proceedings of 4th Aerospace Computer Security Applications Conference*, 1988

[6] Elisa Bertino, Elena Ferrari, Vijayalakshmi Atluri, " A flexible model supporting the specification and enforcement of role-based authorization in workflow management systems" ,*Second ACM Workshop on Role-Based Access Control*, 1997.

[7] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli, " Proposed NIST Standard for Role-Based Access Control" , *ACM Transactions on Information and Systems Security*, Vol.4, No. 3,pp.12-15,2000.

[8] Hyunghyo Lee, Bongnam Noh, " An Integrity Enforcement Application Design and Operation Framework in Role-Based Access Control Systems:A Session-Oriented Approach" , IWSEC,1999.

[9] 지희영," 작업간 비밀성을 보장하는 클래스 기반의 동적 의무 분리 모델" , 통신정보보호학회 논문지, Vol.10, No.2, 2000.

[10] Oh Sejong and Park seog, " Enterprise Model as a Basis of Administration on Role-Based Access Control" , *Proceedings of the third international symposium on Cooperative Database Systems for advanced Applications 2001, IEEE Computer*, 2001.