2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

강제적 접근방식과 역할기반 접근제어 그래프를 기반으로 한 보안모델 설계

박기홍, 김응모
성균관대학교 전기전자및컴퓨터공학과
e-mail : hongyi@ece.skku.ac.kr

Security Model Design based on MAC and RBAC Graph

Ki-Hong Park, Ung-Mo Kim
Dept. of Electric, Electronic and Computer Science,
SungKyunKwan University

요 약

일반적인 운영체제의 보안과 마찬가지로 데이터베이스에서도 보안의 중요성은 강조되고 있다. 다중등급을 가지고 있는 데이터베이스에서 상위등급의 사용자가 사용하는 상위등급 데이터가 하위등급의 사용자가 사용하는 하위등급 데이터로 유입된다면 데이터의 무결성(integrity)이 깨지게 되어 데이터베이스뿐만 아니라 시스템 전체의 보안도 위협받게 된다. 본 연구에서는 대량의 데이터베이스 환경에서 다양한 보안등급을 가지고 있는 사용자가 다양한 등급을 가지고 있는 데이터베이스에 접근할 때 이를 강제적 접근제어(MAC:Mandatory Access Control)와 역할기반 접근제어(RBAC:Role-Based Access Control) 그래프를 이용해 사용자 보안등급에 따른 접근과 상위등급의 데이터가 하위등급으로 유출되지 않도록 이를 효율적으로 관리하고 제어할 수 있는 보안 모델을 제시하는데 중점을 두었다.

1. 서론

최근 네트워크의 발달과 많은 데이터의 증가로 인해 개인이 접근하게 되는 정보량이 기하급수적으로 증가했다. 방대한 정보를 효율적으로 관리하기 위해 데이터베이스의 유지는 필연적이라 할 수 있다. 일반적인 운영체제와 마찬가지로 데이터베이스에서도 보안은 강조되고 있다. 데이터베이스 보안의 목적은 비밀성(secrecy)과 무결성(integrity), 그리고 가용성(availability)이라고 할 수 있다[1]. 첫째, 비밀성은 정보의 부당한 유출을 저지하고 그러한 위협을 검출해 내는 것이다. 둘째, 무결성은 정보에 대한 부당한 수정을 저지하고 데이터의 일관성을 유지하는 것이다. 셋째, 가용성은 보안이 유지된 정보를 실제적으로 사용할 수 있도록 서비스를 하는 것이다.

데이터베이스 보안의 가장 핵심은 적합한 등급을 갖고 있는 사용자가 데이터베이스에 접근할 때 부당한 자료의 유출과 변경이 이루어지지 않으면서 그 사용자에게 허가된 데이터베이스를 효과적으로 사용할 수 있도록 서비스를 제공하는데 있다.

본 논문에서는 강제적 접근제어(MAC:Mandatory Access Control)와 역할기반 접근제어(RBAC:Role-Based Access Control) 그래프를 이용해 보안을 유지하면서 효율적으로 관리하고 제어할 수 있는 보안 모델을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 MAC과 RBAC의 개념을 소개하고, 3장에서는 MAC과 RBAC을 응용한 모델의 설계를 제시하고, 4장에서는 예제 데이터베이스를 통한 Role Graph를 표현하며, 마지막 5장에서는 결론을 기술한다.

2. 관련연구

본 장에서는 기존의 MAC과 RBAC의 특징을 기술한다.

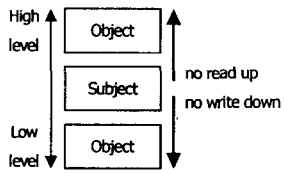
2.1 MAC(Mandatory Access Control)

다중등급을 갖는 데이터베이스 보안에 있어서 접근방식은 크게 두 개의 방식으로 나눌 수 있다. 임의적 접근제어(DAC:Discretionary Access Control) 방식과 강제적 접근제어(MAC:Mandatory Access

Control) 방식이다[1]. DAC은 주체의 신분을 기반으로 하는 접근통제로서 일반적인 상업용 관계형 데이터베이스에서 사용되고 있다. 한편, MAC은 비밀등급 비교에 의한 객체 접근통제방식으로 대표적인 모델은 BLP(Bell-LaPadula) Model[2]을 들 수 있다.

BLP Model은 각 보안등급간에 상위등급의 정보가 하위등급으로 유출되는 것을 막기 위한 모델이다. 그 기본개념은 객체에 접근할 수 있는 일반적인 사용자로 대표할 수 있는 주체(Subject)와 주체가 접근하는 데이터베이스와 같은 객체(Object), 그리고 객체에 할당된 등급(Classification), 주체에 할당된 등급인 접근허가(Clearance)로 나눌 수 있다. 객체와 주체간의 보안등급은 U(Unclassified), C(Classified), S(Secret), TS(Top-Secret)의 4단계로 구분한다. 보안등급은 U에서 TS로 올라 갈수록 높아진다. BLP Model은 두 가지 법칙(그림 1)을 갖는다[2].

- Simple Security Property : Subject S can read object O only if L(S)≥L(O)
- *(star)-Property : Subject S can write object O only if L(S)≤L(O)



(그림 1) BLP Model

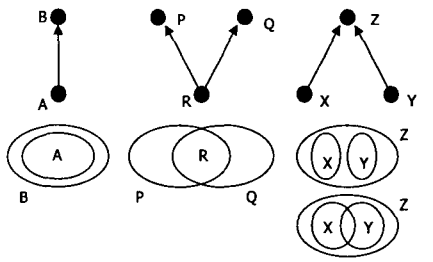
첫째, Simple Security Property는 "no-read up"으로 표현할 수 있는데, 하위등급을 가진 주체는 자신의 등급보다 높은 상위등급의 객체를 read 할 수 없다. 둘째, *-Property는 "no-write down"으로 표현할 수 있는데, 상위등급을 가진 주체는 자신의 등급보다 낮은 등급을 가진 객체에 write 할 수 없다. 만일 하위등급을 가진 주체가 상위등급에 write 할 수 없다면 하위등급은 상위등급에 자신의 등급으로 접근할 수 없는 정보가 있다는 것을 알 수 있으므로 하위등급으로의 정보의 유출이 발생한다. BLP Model은 SeaView Model[3]에서 Polyinstantiation을 허용해 하위등급의 주체가 상위등급의 객체에 write 할 때 상위등급의 주체와 하위등급의 주체가 판별할 수 있는 정보를 구분해 하위등급은 마치 자신이 행한 write가 이상 없이 수행되었음을 느낄 수 있게 한다.

2.2 RBAC(Role-Based Access Control)

RBAC은 시스템에서 방대한 사용자(user)와 데이터 객체(data object), 인가(permission)를 관리하는 방법으로 사용된다[4]. RBAC은 보안모델에서 주체와 객체간 read와 write 동작을 간단하게 도식화함으로써 관리를 용이하게 한다. Role은 Role name과 Privilege로 구성된다.

- Privilege (x,m) : x is the object, m is set of the access mode
- Role(rname,rpset) : rname is the name of role, rpset is set of privileges of the role

RBAC에서 사용자에게 권한부여(Authorization)는 User/Group Authorization, Role/Role Authorization, Role/Privilege Authorization으로 나눌 수 있고 이러한 관계를 통해 사용자는 권한의 투명성을 제공받는다. 지금까지 정리는 결국 Role Graph를 구현하기 위한 것이다. Role 관계는 is-junior 관계로 표현되는 1) Partial Privileges와 common-junior 관계로 표현되는 2) Common Privileges, common-senior로 표현되는 3) Augmented Privileges로 나눈다(그림 2).



(그림 2) Basic Role Relationships

1) Partial Privileges : RoleA⊂RoleB
2) Common Privileges : RoleR is RoleP∩RoleQ
3) Augmented Privilege : (RoleX∪RoleY)⊂RoleZ

위의 세 가지 법칙을 사용해 방향성을 지닌 단선이며, Cycle을 형성하지 않는 Role Graph를 만든다. Role Graph의 구성요소는 다음과 같다.

- MinRole : Minimum Privilege Set or Ø
- MaxRole : Set of every Privilege
- junior : RoleX is junior if RoleX→RoleY
- senior : RoleY is senior if RoleX→RoleY
- A path from MinRole to every Role
- A path from every Role to MaxRole

• path : RoleX→RoleY if RoleX's Privilege Set⊂ RoleY's Privilege Set

Role Graph의 장점은 대용량 데이터베이스 환경에서 이러한 Role Graph를 구현하고 그 형태를 유지, 재구성함으로써 관리자는 탄력성 있게 보안을 유지할 수 있다.

3. MAC과 RBAC을 응용한 모델의 설계

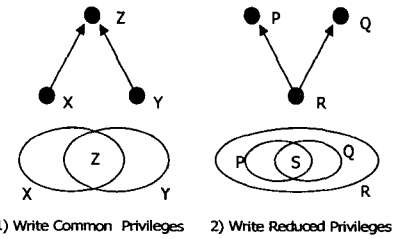
기존의 Role Graph는 MinRole에서 MaxRole로 각 등급을 이루며 설계되었다. read의 경우는 상위등급의 Role이 하위등급의 Role을 포함하면서 Simple Security Property를 만족하지만 *-Security Property를 적용한 write 경우 Role Graph로 표현하는데 어려움이 발생했다. 이런 이유로 lattice of security label을 적용한 LBAC[5]에서는 read, write를 Role Graph로 설계하기 위해, Dual character 구조로 read, write를 구분해서 접근했다. write 경우 1) Liberal *-Property 또는 2) Strict *-Property를 적용해 상위등급의 정보가 하위등급으로 유출되는 것을 막았다[6][7]. 이런 lattice를 적용해 Role Graph를 설계하면 각 등급의 사용자에게 Role Graph level 할당시 각 read, write는 각각 두 그래프로 표현된 후 병합(merge)된다.

1) Liberal *-Property : Subject S can write object O only if L(S)≤L(O)

2) Strict *-Property : Subject S can write object O only if L(S)=L(O)

lattice의 경우 등급을 H(High), L(Low), 그리고 M1(Middle1), M2(Middle2)으로 표현을 해 MAC에서 기본적으로 주장하고 있는 4등급(U, C, S, TS)을 Role Graph로 표현하기 힘들다. 또한 read, write로 구분을 해서 표현할 경우 read의 경우 Role Graph에서 각 객체들이 순차적인 보안등급을 나타내나, write의 경우 각 객체들은 역순의 보안등급으로 표현되었다. 이 경우 보안등급을 가지고 있는 주체에 객체를 할당을 하는데 있어 복잡성이 증대되고, 관리측면에서 쉽게 접근할 수 없다는 단점이 있다.

기존의 MAC의 Simple Security Property와 *-Property를 적용해 상위정보가 하위정보로 유출되는 것은 막으면서, write의 경우는 기존의 Basic Role Relationship에 Extension Role Relationship(그림3)을 첨가해 적용했다. MinRole, MaxRole은 read, write 두 가지 경우로 구분해 설계했다. read의 경우 MaxRole(read), MinRole(read)은 Basic Role Relationship을 적용하고, write의 경우 MaxRole(write), MinRole(write)은 Extension Role Relationship을 적용했다. 결국 MaxRole(write)은 ∅이고, MinRole(write)은 상위등급의 정보가 하위등급으로 유출되지 않도록 각 등급의 사용자에게 Polyinstantiation을 허용하면서 각 등급의 wirte 객체를 포함한다. read, write로 구분된 것을 각각 병합(merge)하여 MaxRole과 MinRole을 만든다. 이렇게 설계된 Role Graph는 각 Role에 접근하는 주체를 기준으로 각 등급의 주체가 할당받은 등급에서 접근할 수 있는 객체가 엄격히 구분된다. 지금까지 내용을 정리하면 다음과 같다.



(그림 3) Extension Role Relationships

1) Write Common Privileges : RoleZ is RoleX∩ RoleY

2) Write Reduced Privileges : (RoleP is subset of RoleR) and (RoleQ is subset of RoleR) and ((RoleP∪RoleQ)⊂RoleR)

• MaxRole(write) : Privilege Set if defined or ∅

• MinRole(write) : Privilege Set of all write Privilege

• MaxRole : MaxRole(read)∪MaxRole(write)

• MinRole : MinRole(read)∪MinRole(write)

• (rname, read.rpset) : RoleA→RoleB only if Basic Role Relationship

• (rname, write.rpset) : RoleA→RoleB only if Extenshion Role Relationship

• (rname, (read.rpset, write.rpset)) : RoleA→ RoleB either Basic Role Relationship or Extenshion Role Relationship

• Role Graph is merged read Role Graph with write Role Graph

4. Role Graph

앞에서 서술한 것을 바탕을 Role Graph를 설계한다. 데이터베이스의 각 table에는 등급이 설정(그림

4)되어 있으며 등급에 맞는 사용자가 접근할 수 있도록, 등급별로 구분한다. 단 테이블에 read, write가 모두 존재할 때는 분리한다(그림 5).

Database

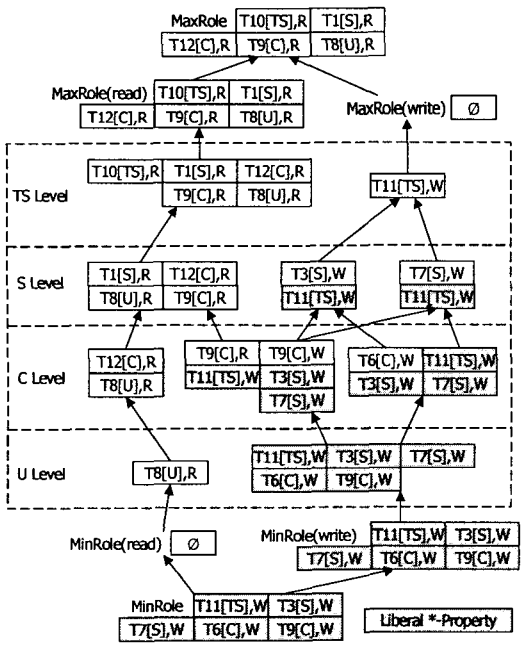
T1[S],R	T8[U],R	T9[C],R/W	T10[TS],R	T12[C],R
T6[C],W	T11[TS],W	T3[S],W	T7[S],W	T : table

(그림 4) Database

T8[U],R	Database		
T6[C],W	T9[C],R	T9[C],W	T12[C],R
T1[S],R	T3[S],W	T7[S],W	
T10[TS],R	T11[TS],W		

(그림 5) 정렬된 Database

관리자는 각 등급별로 포함관계 및 전체적인 구조를 쉽게 파악할 수 있고 Role Graph를 유지, 재구성함으로써 탄력적으로 보안을 유지할 수 있다. Role Graph(그림 6)는 다음과 같다.



(그림 6) Role Graph

5. 결론

데이터베이스에서 데이터의 무결성과 보안을 유지하기 위해 MAC 정책과 Role을 사용해 Role Graph로 표현했다. 상위의 정보가 하위로 부당하게 유출되지 않도록 MAC의 Simple Security Property와 *-Property를 적용해 보안의 누수를 막았다. 또한 Liberal *-Property와 Strict *-Property 모두 Role Graph상에 표현됨을 볼 수 있다(그림 6). 기존 모델과 비교해 각 등급의 주체가 접근할 수 있는 객체의 한계를 명확히 보이고, 그 등급에서 접근될 수 있는 객체를 확인함으로서 관리자가 Role Graph를 탄력적으로 유지, 재구성함으로 데이터베이스 보안관리를 효율적으로 수행할 수 있다.

참고문헌

[1] Silvana Castano, Maria Grazia Fugini, Giancarlo Martella and Pierangela Samarati, Database Security, ACM press, 1995.

[2] D.E. Bell and L.J. Ra Padula. Secure computer System: Unfied Exposition & Multics Interpretation. Technical report, Technical Repory MTIS AD-A023588, MITRE Corporation, 1975.

[3] Teresa F. Lunt, Dorthy E. Denning, Roger R. Schell, Mark Heckman, and William R. Shockley, The SeaView Security Model, IEEE Transation On Software Engineering, VOL. 16, NO. 6, June 1990.

[4] R.S. Sandhu. E.J. coyne, H.L. Feinstein, and C.E. Younam. Role-based access control models. Computer, 29:38-47, Feb. 1996.

[5] R.S. Sandhu. Lattice-based access control models. Computer, 26:9-19, Nov. 1993.

[6] S. Osborn, R. Sandhu and Q. Munawer. Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. ACM Transactions on Information and Systems Security, vol.3, no. 2, 2000.

[7] R.S. Sandhu, Role Hierarchies and Constraints for Lattice-Based Access Controls, ESORICS, 1996.