제20회 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호 (2003. 11)

# 보안 운영체제를 위한 확장된 역할기반 접근통제 기법

신욱*, 이동익*, 김형천**, 강정민**, 이진석**
*광주과학기술원 정보통신공학과
**국가보안기술연구소
e-mail : sunihill@kjist.ac.kr

# Extended Role Based Access Control for Trusted Operating Systems

Shin, Wook*, Lee, Dong-Ik*,
Kim, Hyoung-Chun**, Kang, Jung-Min **, and Lee, Jin-Seok**
*Dept. of Info. & Comm., Kwang-Ju Institute of Science and Technology
**National Security Research Institute

요 약

본 논문에서는 보안 운영체제를 구현함에 있어서, 보다 진보적인 형태의 접근통제를 시행하기 위한 새로운 접근통제 기법에 대하여 설명한다. 새로운 접근통제 기법은 역할 기반 접근통제 기법(RBAC)을 확장하여 구성한다. 또한, 개념의 정확한 표현 및 논리의 정확성 확인을 위하여 정형 기법을 이용, 접근통제 모델을 기술하고 분석한 결과에 대하여 설명한다.

## 1. 서론

보안 운영체제는 신뢰할 수 있는 전산 환경(TCB: Trusted Computing Base)의 구현이다. 보안 운영체제는 기존 운영체제에 보안 커널을 이식하고, 인증, 암호화 등의 보안 서비스를 강화하여 구성한다[1].

보안 운영체제의 동작 기반을 제공하는 보안 커널은 참조모니터(Reference Monitor)의 구현이며, 접근통제 기능을 제공한다. 접근통제를 위한 보안 정책은 크게 임의접근통제(DAC: Discretionary Access Control), 강제접근통제(MAC: Mandatory Access Control), 역할기반 접근통제(RBAC: Role Based Access Control)로 나눌 수 있으며, 이는 보안 운영체제 구현을 분류하는 기준으로 사용되기도 한다.

접근통제는 시스템 내에서 매순간 발생하는 접근(access)이 적절한지를 판단하는 과정이다. 참조모니터는 접근 발생 시, 접근 주체(access subject)와 객체(access object)로부터 접근 결정(access decision)에 필요한 정보를 추출하며, 보안 규칙에 위배되지 않는지를 판단한다[2]. 이러한 판단과정은 접근이 발생할 때 마다 반복된다.

그러나, 기존의 접근통제 기능에는 한계가 있다. 접근 결정에 필요한 정보는 매번 새롭게 얻어지며, 접근 행위 간 연관관계는 고려되지 않아왔다. 즉, 접근 결정을 위해 사용되는 정보들은 접근 결정이 내려진 후, 유효성을 상실한다. 따라서, 공격이 합법적인 연산의 조합을 통해 이루어지는 경우, 접근통제 서비스는 이를 제대로 제어하지 못한다.

이러한 공격을 탐지하고 제한하기 위해서는 각 접근행위 간 연관관계를 추가적으로 고려해야 한다. 즉, 접근통제를 위한 시야(view)를 확대하여, 발생하는 일련의 접근 행위 집합이 공격에 해당하는지를 검사하고 현재 접근이 유효한지를 판단해야 한다.

본 논문에서는 이를 위해 최근 보안 운영체제 구현에 활발히 적용되고 있는 접근통제 기법인 RBAC 을 확장하여 새로운 접근통제 기법을 제안하고자 한다.

논문의 구성은 다음과 같다. 2 장에서는 확장된 접근통제의 개념에 대하여 설명하고, 3 장에서는 정형적으로 모델링한 결과를 보이며, 4 장에서 제안한 기법을

간략히 분석하고 5 장에서 결론을 맺는다.

## 2. 확장된 RBAC

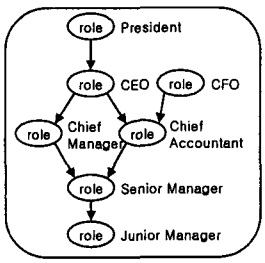
기존 접근통제 기법으로 적절히 대응할 수 없었던 공격들 중, 일례를 TOCTTOU(Time-Of-Check-To-Time-Of-Use)[3] 취약점을 이용하는 경쟁 조건(race condition) 공격에서 찾아볼 수 있다. Sendmail 프로그램의 특정버전은 수신 메일을 저장하기 위한 임시파일을 다른 파일로 대체하는 것이 가능하다. 이를 이용하면, 중요파일의 내용의 불법 변조가 가능하다[4]. 공격 과정에서 핵심적인 역할을 담당하는 연산들은 프로세스 실행 및 링크 설정, 해제 연산들이며, 일반적으로 허용된 연산들이다. 기존 운영체제의 DAC 기반 접근통제 메커니즘은 개개 연산이 보안 규칙을 위해하지 않는다고 판단한다.

이러한 공격에 대응하는 방법은 IDS 등의 감시 프로그램을 사용하는 방법 등 여러 가지가 있지만[3], 보안 운영체제의 커널 수준 접근통제 기능을 제어할 경우, 우회할 수 없는 원천적 제어 메커니즘을 마련할 수 있으므로 효과적이다.

RBAC 을 확장하여 이러한 접근통제를 실현할 수 있다. 앞서 언급한 바 있듯, RBAC 은 최근 보안 운영체제로 활발히 적용되고 있는 접근통제 기법이다. 기존 범용 유닉스 호환 운영체제들은 임의접근통제(DAC: Disretionary Access Control)를 기반으로 운영되어 왔으나, 정보 흐름 통제의 불가로 인한 각종 보안 취약성이 지적됨에 따라, 강제접근통제(MAC: Mandatory Access Control)를 적용하여 보안성을 강화하려는 일련의 노력이 있었다[5,6,7]. 그러나, MAC 의 지나친 제약으로 인한 실용의 어려움으로 인하여, 최근에는 역할기반접근통제(RBAC: Role Based Access Control)를 적용하여 DAC 과 MAC 의 단점을 보완하고자 하는 노력이 진행되고 있다[8,9,10].

RBAC 의 도입으로 인한 장점에는 여러 가지가 있지만, 주목할 만한 것은 역할을 통해 제약조건을 부여하기가 용이하다는 것이다. RBAC 의 주 구성요소는 사용자, 역할, 권한이다[11,12]. 기존 접근통제 방식과 구별되는 RBAC 의 가장 큰 특징은 접근 주체와 객체 사이에 역할이라는 개체를 도입하여 주체와 객체의 추상화(abstraction)를 진행하고, 관리적 편의를 도모함에 있다. 가령 의료 시스템에 '의사', '간호사' 등의 역할을 정의하여 접근통제를 행할 경우 사용자 배정이나 권한 관리 상의 편의를 도모할 수 있다 [11,12,13]. 사용자 조직의 환경 정보(context information)를 기초로 정의되는 이러한 역할들은 사용자 추상(user abstraction)의 결과이다. 한편, '계정관리', '감사'와 같이, 몇몇 권한의 집합으로 시스템 내에 정의된 기능(function)을 표현할 수도 있다. 이러한 역할들은 객체 추상(object abstraction)의 결과물이다. 역할들은 때때로 계층 구조를 갖거나, 제약조건을 가짐으로써, 목적 시스템의 보안 정책의 상세한 표현을 돕는다.

접근 통제의 기능 확장은 역할의 추상화기능으로부터 기인한다. 역할의 객체 추상 과정에서, 역할에 배정된 객체 접근 권한에 순서 정보 등을 삽입하여 연




[그림 1]


[그림 2]

속된 접근행위를 표현하는 것이 가능하다.

현재 RBAC 의 접근객체, 즉 권한(permission)은 시스템 객체들과, 이에 대한 접근연산모드로부터 유도된다[13]. 예를 들어, 파일에 대한 읽기, 쓰기, 수정, 링크 설정, 링크 해제 등의 연산으로부터 각각의 권한이 유도되었다고 가정하자. 앞에서 언급한 레이스 컨디션 공격을 제지하기 위하여 그림 2 와 같은 권한 수행 절차를 하나의 역할로 표현할 수 있다.

이를 위해 RBAC 개념의 확장이 필요하다. 먼저, 권한의 집합인 역할 개체에 순서와 반복을 표현하기 위한 정보를 삽입하여야 한다. 이러한 정보의 표현 후에는 하나의 역할이 순서화된 연산 집합, 즉 절차(procedure)를 의미할 수 있게 된다. 또한, 특정 권한집합이 공격을 대표할 수 있도록 부정권한(negative permission)[1]임을 표현하는 정보를 삽입하여야 한다. 부정(negative)에 해당하는 역할은 공격으로 인식되고, 권한들이 부정적 절차에 따라 수행 완료되기 전에 접근통제는 이를 거부하게 된다. 이 밖에 연산 실행의 시각, 시간 정보, 객체의 소유자 등에 관한 정보 등을 추가하여 보다 정밀한 통제를 시행할 수 있다.

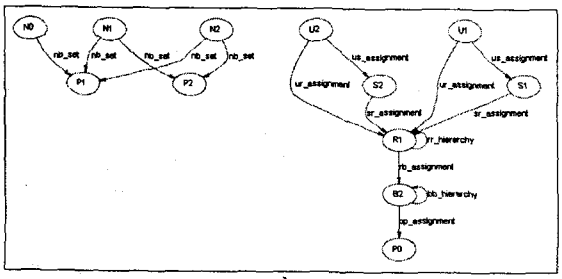
역할의 개념을 이렇게 확장할 경우, 객체 추상에 해당하는 역할들은 사용자 추상을 위한 역할들과 의미와 표현 상 확연히 구분된다.

역할은 접근 주체이면서 동시에 접근 객체로 해석될 수 있는 개체이지만, 여러 RBAC 관련 연구들에 나타난 예제를 살펴보면, 역할 계층에 반영되는 주된 정보는 사용자 조직으로부터 추출되어 왔음을 알 수 있다[11,12,14]. 그림 1 은 사용자 조직을 반영하여 구성한 역할 계층의 전형이다[15]. 따라서, 객체 추상을 위한 정보들을 사용자 추상 개체에 덧붙일 경우 다음과 같은 단점이 발생할 여지가 있다.

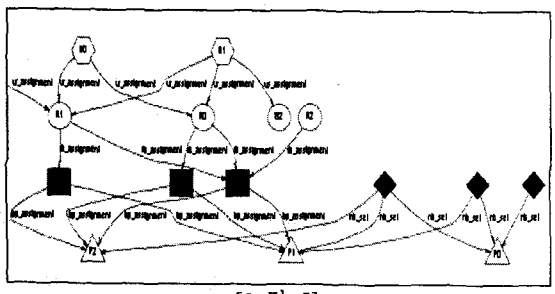
- 구현상의 오버헤드: 연산 수행 순서, 시간, 소유권 등의 정보는 사용자 추상과정에서는 불필요한 정보이다. '의사', '간호사' 등의 역할에 불필요한 필드를 추가하게 되면 공간의 낭비가 발생하며, 이러한 자료구조를 초기화하기 위해 불필요한 루틴들을 작성하게 될 가능성이 있다.

```
model RBBAC {
        domain{ Users, Roles, Sessions, Behaviors, Permissions, NBehaviors }
        state{
                ur_assignment: Users -> Roles
                rb_assignment: Roles -> Behaviors
                bp_assignment: Behaviors -> Permissions
                us_assignment: Users! -> Sessions
                sr_assignment: Sessions -> Roles+
                rr_hierarchy: Roles -> Roles
                bb_hierarchy: Behaviors -> Behaviors
                nb_set: NBehaviors -> Permissions
        }
}
```

[그림 3]



[그림 4]



[그림 5]

- 의미적 괴리: 역할 이라는 같은 추상 계층 내에, 사용자 추상의 결과물과 객체 추상의 결과물이 혼재하는 것은 개념적으로 부적절하다. 또한, 주체로부터 객체로 이어지는 접근통제 개체간의 관계(relation)는 부분순서(partial order)를 가지게 마련인데, 단일 계층 내에 혼재된 주체 및 객체의 추상 개체는 보안 규칙 생성 시 객체가 주체를 상속받는 형태의 순서 역전 상태를 유도할 수 있다. 이러한 순서 역전은 사이클을 생성하여 역할의 상속계층을 붕괴시킬 수 있다.

이러한 단점을 피하기 위하여, 사용자의 추상계층과 시스템 자원의 추상계층을 분리시키는 것이 합리적이다. 기존 RBAC 의 역할계층은 사용자의 추상화만을 담당하도록 하고, 연산의 추상화를 담당하던 역할들을 새롭게 행위(behavior)라고 명명, 역할과 별도로 정의하는 형태의 접근통제 기법을 가정해 볼 수 있다.

확장된 기법의 주 요소는 사용자, 역할, 행위, 권한의 네가지로 구성된다. 역할은 접근 주체의 추상화를 행위는 객체의 추상화를 담당하는 계층이며, 주체의 속성과 객체의 속성에 따라 별도의 제약조건을 부여할 수 있다. 역할에는 사용자 도메인, 접근 시간 등의 제약을 부여할 수 있으며, 객체에는 연산수행 순서, 실행시간, 소유권 등의 조건을 부가할 수 있다. 다음 장에서는 이러한 기법을 수학적 표현을 빌려 모델링한 결과에 대하여 설명한다.

## 3. 확장된 RBAC 의 정형 모델

이 장에서는 확장된 RBAC 을 정형적으로 표현한 결과를 제시한다. RBAC 의 확장 모델은 집합 표현을 빌려 기술하는 것이 일반적이나, 이러한 기술의 기본적 형태는 기존 연구[16]를 통해 이미 제시하였고, 지면이 한정적이므로 본 논문에서는 생략한다. 단, 여기서는 이미 기술한 바 있는 정형 표현을 Schaad[17]의 연구에서 사용된 바 있는 Alloy 를 사용하여 명세한 결과에 대해 언급하기로 한다. Alloy 는 배우기 쉽고, 사용하기 쉬우면서도 구조적인 언어체계를 가진 정형 언어이다. 또한 정형 표현 기술된 시스템의 마이크로 모델을 분석해주는 도구를 지원하고 있다[18].

그림 3 은 확장된 RBAC 의 기본적인 구조를 Alloy 로 표현한 결과의 일부이다. Alloy 는 확장된 RBAC 의 각 요소와 요소간의 가능한 관계를 그림 4 와 같이 표현해준다. 또한, 그림 5 는 기술된 확장 RBAC 모델의 분석 결과 도식 중의 일부로서, 붉은 마름모로 표시된 실행 금지된 행위의 집합이 푸른 사각형으로 표시된 실행 허가된 행위와 동치관계인 경우가 없도록, 즉 시스템 공격에 해당하는 부정적 권한 집합을 사용자가 실행할 수 없도록 불변식(invariant)이 기술되어 있음을 확인할 수 있다.

## 4. 분석

확장된 접근통제 기법은 기존의 RBAC 과는 구별되는 기법이다. 기존의 RBAC 에서의 역할이란 접근주체로도, 객체로도 해석될 수 있는 중립적인 존재였으며, 주체와 객체간의 매핑(mapping)이 형상화된 개념이었다. 그러나 확장된 기법의 역할은 주체를 대표하는 개념, 행위는 객체를 대표하는 개념이다. 보안 규칙의 핵심이라 할 수 있는 주체와 객체간의 매핑은 기존에는 역할로 개체(entity)화 되었으나, 확장된 기법에서는 여전히 역할과 행위간 관계(relation)로 존재하며 개체화 되지 않는다. 시각을 달리하자면 역할은 그룹으로, 행위는 함수(function)으로도 해석될 수 있다. 이러한 변화로 인하여 기존 RBAC 이 제공하던 편의가 얼마나 줄어드는 지에 대한 정성적, 정량적 분석은 필요한 것이지만, 본 논문에서는 논외로 하였으며 향후 연구를 통해 규명해 보고자 한다.

새롭게 확장된 RBAC 기법은 최소권한(Least privilege)의 원칙과 의무 분리(Separation of duty)의 원칙을 보다 견고히 준수할 수 있도록 돕는다. 가령, 특정 시스템에 {A, B, C, D}라는 권한 집합이 존재하고 있다면, 기존의 접근통제 기법은 집합 전체를 해당 구성원에게 허용한다. 즉, 연산 집합이 다양한 순서로 실행될 수 있음에도 불구하고, 모든 순서의 실행을 허가하므로, 특정 작업이 (A, C, B, D)로 수행된다는 것을 예상한다 하더라도, (B, D, C, A)로 수행되는 일련의 연산을 역시 허용하는 셈이다. 그러나, 확장된 RBAC 은

권한의 순서화된 기술을 지원하므로, 최소 권한 원칙을 보다 정확히 준수할 수 있도록 한다. 또한, 특정 연산 순서 (A, C, B, D)와 (B, D, C, A)가 의무분리 대상일 경우, 기존의 접근통제 기법과는 달리 보다 구체적인 의무 분리를 행할 수 있다.

## 5. 결론 및 향후 연구

본 논문에서는 보다 진보적인 접근통제의 실현을 위하여 유연한 접근통제 기법인 RBAC 을 확장, 새로운 접근통제 기법을 제안하였으며, 정형적으로 모델링하여 새 기법의 특성과 논리적 표현의 정확성을 확인하였다. 새롭게 제안한 접근통제 기법의 특징은 시스템 자원을 추상화 할 수 있는 독자적인 통로를 마련하는 것이며, 특히 추상화 과정에 절차적 제약을 도입하여 접근통제를 연속적(continuous) 형태로 확장한 것에 있다. 절차적 제약은 부정적인 시스템 연산을 정의하고, 접근통제 시스템이 해킹 등 시스템 공격의 실행을 보다 정확히 구별하고 막을 수 있도록 한다. 이러한 기능은 침입탐지시스템(IDS)에서 볼 수 있는 기능과 유사한데, IDS 가 어플리케이션 수준(level)에서 동작하는 반면, 접근통제는 우회할 수 없는 운영체제 커널 수준에서 동작하므로 제어에 유리하다. 절차적 제약을 자원 추상화 과정에 도입함으로써 얻을 수 있는 또 하나의 장점은 최소권한의 원칙과 의무분리 원칙을 보다 정확히 준수할 수 있다는 것이다. 기존의 권한 집합에는 절차의 개념이 존재하지 않으므로, 개개 권한의 수행 순서를 고려하지 않았으나, 새 기법에서는 지정된 순서에 따른 권한 실행만을 허용하는 것이 가능하다. 단, 논문에서 언급한 현재의 정형 모델은 절차를 모델링 하지 않고 있다. 이는 나머지 모델의 기술이 정형 기법 중 증명 기반 기법을 이용하여 기술되었으나, 절차적 제약부분은 상태 기계 기반 기법을 이용해야 하기 때문이다. 따라서, 향후연구로 두 정형기법을 적절히 활용하여 정형 모델을 완성한 후, 실제 시스템에 구현하고자 한다.

## 참고문헌


[1] D. Gollmann, "Computer Security," John Wiley & SONS 1999.

[2] ITU-T SG/7 & Working Parties, "Final text for recommendation X.812 Information Technology-Open Systems interconnection Security framework for open systems: Access control framework," 1995.

[3] M. Bishop and M. Dilger, "Checking for Race Conditions in File Access," Computing Systems 9(2), pp. 131-152, Spring 1996,

[4] 8LGM, "Advisory 20", [8GM]-Advisory-20.UNIX.SunOS-sendmailV5.1-Aug-1995.README

[5] "UNICOS Multilevel Security (MLS) Feature User's Guide," SG-2111 10.0, Cray Research, Inc. 1990.

[6] M. Branstad, H. Tajalli, and F. Mayer, "Security issues of the Trusted Mach system," Proc. of 4th Aerospace Computer Security Applications Conference, pp. 362-367, 1998.

[7] Flask: http://www.cs.utah.edu/flux/fluke

[8] P. Loscocco, S. Smalley, "Integrating Flexible Support for Security Policies into the Linux Operating System," Proc. of the FREENIX Track: 2001 USENIX Annual Technical Conference (FREENIX '01).

[9] A. Ott, "The Rule Set Based Access Control (RSBAC) Linux Kernel Security Extension," 8th Int. Linux Kongress, Enschede 2001.

[10] Trusted Solaris: http://wwws.sun.com/software/solaris/trustedsolaris/index.html

[11] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," IEEE Computer, Vol. 29, No. 2 1996.

[12] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," ACM Transactions on Information and Systems Security, Vol. 4, No. 3 2001.

[13]D. Ferraiolo, J. Cugini, R. Kuhn, "Role Based Access Control: Features and Motivations," Proceedings, Annual Computer Security Applications Conference, IEEE Computer Society Press, 1995.

[14] Moffett, J. D, "Control Principles and Role Hierarchies," 3rd ACM Workshop on Role Based Access Control (RBAC), George Mason University, Fairfax, VA, 22-23 October 1998.

[15] M. Koch, L.V. Mancini and F. P. Presicce, "A Graph-Based Formalism for RBAC," ACM Transactions on Information and System Security, Vol. 5, No. 3, pp. 332-365, August 2002.

[16] 신욱, 이동익, 윤석환, "워크플로우 응용을 위한 이동 에이전트 시스템에의 역할-행위 기반 접근통제 적용", 정보보호학회논문지, 제 10 권, 3 호, pp. 11-27, 2000 년 9 월.

[17] A. Schaad and J. Moffett, "A Lightweight Approach to Specification and Analysis of Role-based Access Control Extensions," 7th ACM Symposium on Access Control Models and Technologies (SACMAT 2002), Jun. 2002.

[18] Alloy: http://sdg.lcs.mit.edu/alloy/