RBAC 모델 기반의 보안 관리방안

황승노° 류기열 예홍진
아주대학교 정보통신전문대학원

Security Management Method based on RBAC models

Seungro Hwang° Kiyeol Ryu Hongjin Yeh
Graduate School of Information and Communication, Ajou University

요 약

인터넷에 노출되어 있는 시스템 혹은 내부 네트워크에 대한 해커들의 공격은 날이 갈수록 심각한 상태에 이르고 있으며 이에 따라 이를 방어하기 위한 기법들에 대한 개발도 활발한 상태이다. 아무리 우수한 보안 시스템이라도 이를 어떻게 관리하고 사용하는가에 따라 그 효과가 크게 변한다. 본 논문은 RBAC 모델을 보안 관리에 적용시켜 보안 개체들 간의 효율적 역할 분담 및 상호 견제를 통해 보안 관리 체계를 개선하는 방안을 제안한다.

1. 서 론

일반적으로 보안 시스템은 동일 조직 내의 보안 담당자 혹은 보안 담당 부서 직원들에 의해 운영되고 관리된다. 즉 '조직 내부'의 '인간'에 의해서만 관리된다. 그러나 '조직 내부' 라는 요소와 '인간'이라는 요소는 다음과 같은 보안 관리의 허점을 가지고 있다.

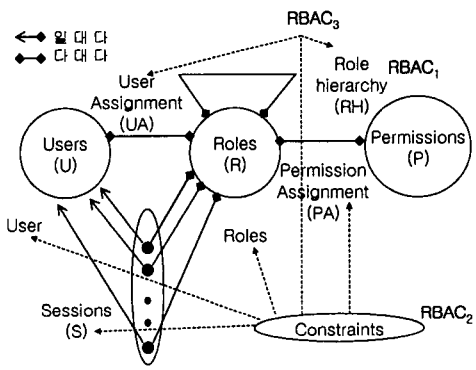
1) 조직 내부 : 이 요소는 묵시적으로 보안 시스템의 업그레이드가 실시간으로 이루어 질 수 없음을 의미한다. 즉, 보안 시스템을 가장 최신 버전으로 업그레이드 할 수 있는 존재는 시스템 공급자이지만 보안 시스템이 조직 내부의 인물에 의해 전적으로 관리되는 환경에서는 그 인물을 통한 시스템의 업그레이드만이 가능하다. 따라서 시스템의 실시간 업그레이드가 이루어질 수 없다.

2) 인간 : 이 요소의 미래의 상태를 정확히 예측할 수 없다. 즉, 시스템 관리자가 인사에 불만을 품고 악의적으로 보안 시스템을 파괴 또는 오용할 수 있다.

1) , 2) 와 같은 허점을 보완하기 위해 보안 관리 주체에 시스템 공급자와 보안 시스템 자신을 포함 시킬 필요가 있다. 이와 같은 경우 기존의 보안 관리 주체와 새로 도입되는 보안 관리 주체들 간의 상호 관계 , 또 각 관리 주체들의 일반 시스템 자원과 보안 관련 시스템 자원의 사용 한계 등을 명확히 정의하기 위해서 각 주체간의 상호 관계 (role hierarchy) , 각 주체와 자원 사용범위 할당에 관련하여 다양한 적용을 제공하는 역할 기반의 접근 제어 모델 (Role-Based Acess Control)을 사용하는 것이 유리하다.

2. RBAC (Role-Based Access Control)

RBAC이란 역할에 기반을 두고 사용자의 시스템 자원에 대한 접근을 제어하는 기법을 말하는 것으로서 수년 전부터 최근에 이르기까지 활발한 연구가 진행되고 있다. 최초 $RBAC_0$ 모델에서 $RBAC_1$, $RBAC_2$, $RBAC_3$ 모델까지 발전되어왔다. [1]



[그림1] RBAC 모델

[그림1]은 $RBAC_0$ 에서 $RBAC_3$까지의 모델을 종합적으로 보여주고 있다. $RBAC_0$ 모델은 아래와 같이 4개의 개체로서 구성된다.

Users(U) : 시스템의 사용자로서 대개 사람을 나타냄

Roles(R) : 조직내의 직급을 나타내며 고유의 권한과 의무를 갖음

Permissions(P) : 시스템 자원에 대한 접근 권한

Sessions(S) : 한 명의 사용자가 하나의 작업을 수행하기위해 자신에게 부여된 1개 이상의 role을 활성화 시킨 상태

$RBAC_1$ 모델은 $RBAC_0$ 모델에 role hierarchy 개념을 추가하여 조직의 권한과 의무 계통을 반영하는 role 간의 구조 설정 수단으로 사용한다. 상위 role은 하위 role의 권한을 상속하여 사용할 수 있다. $RBAC_2$ 모델은 추가로 constraints 를 도입한다. constraints 개념은 RBAC 모델 각 개체에 제약 조건을 부여할 수 있는 것으로서 이를 적용한 한 예로는 동일 사용자가 서로 배타적 role 을 할당 받을 수 없다는 separation of duty가 있다. $RBAC_3$ 모델은 $RBAC_0$, $RBAC_1$, $RBAC_2$ 모델을 통합한 모델이다.

3. RBAC 기반의 보안 관리방안

현재의 보안 시스템은 보안 관리자 혹은 관리 그룹 독자적으로 관리되어진다. 따라서 보안 프로그램 업그레이드 시에는 보안 업체 직원에 의해 오프 라인 으로 서비스를 받거나 관리자가 업체 홈페이지를 접속해 패치 프로그램을 다운로드 받아 프로그램을 업그레이드하게 된다. 이러한 방식은 하루가 다르개 새로운 해킹 기법이 개발되어 인터넷에 접속된 시스템을 공격하는 현실을 감안해 볼 때 신속한 방어를 하는데 있어 큰 단점으로 작용한다. 또한 보안 시스템의 관리자에 의한 보안 시스템의 악의적 수정도 고려되어져야 한다. 따라서 이를 방어하기 위해 시스템 스스로가 방어적 능력을 보유할 필요가 있다.

3.1 각 개체의 원소

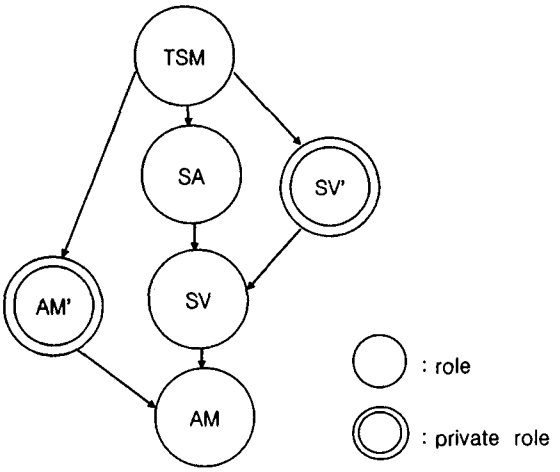
RBAC 모델을 보안관리에 적용함에 있어 각 개체의 원소를 아래와 같이 사용한다.

Users(U) : 보안 담당 직원 , 시스템 공급자 , 시스템 모듈

Roles(R) : 최고 보안 관리자 (Top Security Manager : TSM) , 보안 관리자 (Security Administrator : SA) , 보안 시스템 공급자 (System Vender : SV) , 시스템 자체 방어 모듈 (Autonomous Module : AM)

Permissions(P) : {Operations} * {Objects}

3.2 Role Hierarchy (RH)의 구성



[그림 2] Role Hierarchy (RH) 구성

[그림 2] 와 같이 Role Hierarchy를 구성하여 각 role 간의 견제 기능을 도입한다. 그림에서 SA는 SV와 AM의 기본적인 권한을 상속받는다. 즉 사람의 참여가 필수적으로 요구되지 않는 보안 시스템 관리 기능은 AM이 모두 수행하며 그 기능은 SV , SA도 상속 받아 AM의 에러 발생시 SV 혹은 SA에 의해 수행될 수 있다.

예를 들어 각 센서들로부터 입력되는 시그널을 분석하여 access control table의 수정을 AM이 수행할 수 있다. 그러나 AM이 제 기능을 제대로 수행하지 못하는

경우 이 기능은 SV 혹은 SA에 의해서도 수행되어질 수 있다.

AM'은 AM의 private role 로서 SV와 SA에게는 그 권한이 상속되어지지 않으며 오직 최고 보안 관리자인 TSM에게만 그 권한이 상속된다. 그리하여 AM은 보안 관리의 한 주체인 SV와 SA의 악의적 조작을 견제할 수 있는 능력을 갖게 된다.

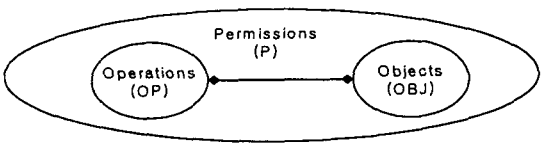
예를 들어 SV 혹은 SA가 보안 시스템에 치명적일 수 있는 성분(factor) 값을 수정한 경우 AM'은 SV 혹은 SA의 이러한 행위에 의한 결과를 일정시간 동안 적용되는 것을 지연 시키면서 TSM 에게 경고 신호를 발송한다. [3] SV의 private role인 SV'에 대해서도 같은 맥락의 설명이 가능하며 AM'과 SV'의 권한은 TSM에 의해서만 상속되어진다.

3.3 Separation Of Duty (SOD) Constraint 설정

동일한 사람이 2개 이상의 conflicting role을 취득할 경우 3.2절에서 설정한 role hierarchy는 그 의미를 잃게 된다. 즉, 동일인이 SA role 과 SV' role, 혹은 SA role 과 AM' role을 동시에 취득한다면 보안 관리 주체 간의 상호 견제는 그 기능을 상실하게 된다. 따라서 이와 같은 conflicting role set에 대해서는 SOD constraint를 설정하며 이것의 구현을 위하여 RCL2000을 사용할 수 있다. [2]

3.4 system 자기 방어 기능 부여

3.2절에서 설명한 AM'의 고유 권한의 구체적 구현 방안으로 [그림 3]의 형태로 제시한 Permission 개체를 고려해 보자. Operation (OP) 개체와 Object (OBJ) 개체로 세분된 Permission의 개수는 {OP} * {OBJ} 가 된다. 여기서 $OP_i \times OBJ_i$ Permission 이 security critical 한 명령 (P_i) 이라 가정하고 AM' role을 enable 시키는 function call을 sequence operation으로 필히 수행 시키는 routine을 첨가한다. 만일 SV 나 SA 가 P_i을 수행하면 AM' role은 enable 되어 자체 방어 기능을 수행하게 된다. [3] , [4]



[그림 3] Permission 개체의 구성

4. 결론 및 향후 연구 과제

본 논문은 보안 시스템 공급자와 시스템 자체를 시스템 모듈의 형태로서 기존의 보안 관리주체에 추가로 도입하였으며 보안관리 체계에 RBAC 기법을 도입하여 이를 통해 보안 시스템의 신속하고 정확한 upgrade를 이루며 각 보안 주체간의 상호 견제를 통하여 내부자의 보안 시스템에 대한 악의적 조작을 방지하는 관리 방안을 제안 하였다. 이 방안을 기초로 한 보안 관리체계의 구체적 모델화와 실제 시스템에 적용을 통한 성능 분석이 향후 과제로 남는다.

참 고 문 헌

[1] R. S. Sandhu, "Role-Based Access Control Models", IEEE Computer, pp. 38-47, Feb., 1996

[2] G. J. Ahn and R. S. Sandhu, "Role-Based Authorization Constraints Specification", ACM Transactions on Information and System Security, Vol. 3, No. 4, pp207-226, Nov., 2000

[3] J. B. D. Joshi, E. Bertino and A. Ghafoor, "Temporal Hierarchies and Inheritance Semantics for GTRBAC", In Proceedings of SACMAT'02, California, USA, Jun. 3-4, 2002

[4] R. S. Sandhu, "Role Activation Hierarchies", In Proceedings of 3rd ACM Workshop on Role-Based Access Control, Virginia, USA, Oct. 22-23, 1998