리눅스 보안 모듈 기반의 확장된 역할 기반 접근 제어 보안 시스템 설계

박신혜[0] 예홍진 김동규
아주대학교 정보 통신 전문 대학원
(sinne[0], hjyeh, dkkim)@ajou.ac.kr

The Design of LSM-Based Extended RBAC Security System

Sihn-hye Park[0] Hong-jin Yeh Dong-Kyoo Kim
Dept. of Information Communication Engineering GSIC AJOU

요 약

오늘날의 규모가 매우 크고 복잡한 시스템 및 네트워크 환경 하에서 컴퓨터 시스템의 보안 관리는 매우 중요하다. 역할 기반 접근 제어 (RBAC: Role-Based Access Control)는 시스템 상의 역할을 기반으로 하는 접근 제어 메커니즘으로 복잡한 접근 정책을 기술하고, 시스템 관리상의 에러와 비용을 줄일 수 있다. 본 논문은 리눅스 커널에 보안 강화를 지원하는 리눅스 보안 모듈(LSM: Liunx Security Module) 프레임워크 상에서 확장된 역할 기반 접근 제어 보안 시스템 (LSM-Based Extended RBAC Security Module)을 제안한다. 본 고에서 제안된 시스템은 보안의 강화를 위하여 원 타임 패스워드 (One-Time Password)의 강화된 인증 방식과 부분적 다중 계층 보안 (Partial Multi-Level Security), 임의적 접근 제어 (Discretionary Access Control) 및 감사 정보를 통한 보안 정책 오류 검사 및 대응 (Security Policy Validation and Response) 기능을 지원한다.

1. 서 론

오늘날의 복잡하고 규모가 큰 시스템 및 네트워크 환경에서 보안 관리는 매우 중요하다. 컴퓨터 시스템 보안을 위한 대표적인 방법으로 시스템 자원에 대한 접근을 제어하는 접근 제어 방식이 사용되며, 이러한 접근 제어 방식에는 임의적 접근 제어, 강제적 접근 제어 및 역할 기반 접근 제어 등이 있다. 특히 역할 기반 접근 제어는 대규모 컴퓨팅 환경에서의 복잡도를 줄이고, 보안 관리의 비용을 줄일 수 있어 다양한 컴퓨터 환경에서 설계 및 구현되고 있다[1]. 역할 기반 접근 제어는 역할을 생성하고, 그 역할에 권한을 부여한 후, 사용자를 할당하는 방식으로 관리된다[2]. 따라서, 역할 기반 접근 제어는 보안 시스템을 구현하고자 하는 기업 구조에 가장 밀접한 보안 레벨을 통하여 관리할 수 있다[3][4]. 또한 역할 기반 제어를 통하여 상호 배타적인 역할(mutually exclusive roles)과 역할 계층 관계(role hierarchy) 등 현실에서 일어날 수 있는 복잡한 문제들을 해결하는 보안 시스템을 구현할 수 있어, 보안 문제를 더욱 쉽게 관리할 수 있다[5].

본 논문은 리눅스 커널에 보안 강화를 지원하는 리눅스 보안 모듈(LSM: Liunx Security Module) 프레임워크 상에서 확장된 역할 기반 접근 제어 보안 모듈(LSM-Based Extended RBAC Security Module)을 제안한다. 본 고에서 제안된 리눅스 보안 모듈 프레임워크 상에서의 확장된 역할 기반 접근 제어 보안 시스템은 보안의 강화를 위하여 원 타임 패스워드(One-Time Password)의 강화된 인증 방식과 부분적 다중 계층 보안 (Partial Multi-Level Security), 임의적 접근 제어 (Discretionary Access Control) 및 감사 정보를 통한 보안 정책 오류 검사 및 대응(Security Policy Validation and Response) 기능을 지원한다.

2. 관련연구

시스템 보안을 위한 접근 제어 방식에는 여러 가지가 있으나, 그 중 대표적인 방식으로 임의적 접근 제어(DAC: Discretionary Access Control), 강제적 접근 제어(MAC: Mandatory Access Control), 역할 기반 접근 제어(RBAC: Role-Based Access Control) 방식이 있다[11].

기존의 유닉스(UNIX) 운영체제는 대부분 임의적 접근 제어 방식을 사용한다[11]. 임의적 접근 제어는 한 사용자가 사용자 소유의 파일, 사용자가 사용하는 프로그램 그리고 사용자의 권한으로 실행되는 프로그램에 대하여 완전한 제어를 할 수 있도록 한다. 이 방식은 사용자가 자신의 오브젝트에 대하여 접근 권한을 줄 수 있으므로 응용 프로그램의 실행에 있어서 시스템 보안이 깨질 수 있다[11]. 강제적 접근 제어는 관리자가 엄격한 보안 접근 제어 목록을 정의한다. 강제적 접근 제어를 사용하는 시스템은 매우 엄격한 정책 하에 동작한다. 유닉스(UNIX) 시스템에서 강제적 접근 제어를 구현하기 위해서는 시스템을 사용하는 모든 사용자에 대하여 보안 정책을 설정하여야 하기 때문에 정책의 규모가 매우 커지는 단점이 있다. 이러한 작업을 쉽게 하기 위하여 역할 기반 접근 제어를 사용할 수 있다. 역할 기반 접근 제어는 역할을 통하여 시스템 자원에 대한 접근을 제어한다. 이러한 접근 제어 방식은 역할과 주체와의 관계, 역할과 권한과의 관계로 표현된다는 것이 특징이다. 즉, 사용자들이 시스템 자원에 임의적으로 접근을 하지 않고, 역할에 대한 권한이 설정되고 사용자들이 그 역할의 구성원이 된다[1][7]. 이 개념은 특정 보호 정책의 명세와 강화에 있어서 매우 유연하여 인증 관리를 손쉽게 한다[1]. 사용자들은 그들의 책임 및 자격에 따라 특정 역할의 구성원이 될 수 있으며 접근 구조의 수정 없이 이 역할에서 저 역할로 쉽게 재할당될 수 있다[1]. 역할은 새로운 응용 프로그램과 작업 내용들이 병합됨으로써 새로운 권한을 얻을 수 있으며, 권한들은

필요에 따라 역할들로부터 호출된다[1].

리눅스 보안 모듈(LSM: Linux Security Module)은 리눅스 커널에 보안 모듈을 제공한 프레임워크를 설계하고 구현한 프로젝트로 보안 강화 리눅스의 기반이 된다[10]. 보안 강화 리눅스(SELinux : Security-Enhanced Linux)는 리눅스 보안 모듈 프레임워크 상에서 강제적 접근 제어 모델을 구현한 것이다. 보안 강화 리눅스 시스템의 관리자는 어떠한 프로그램이 어떠한 파일에 접근 할 수 있는 지의 여부를 정의하는 시스템 상의 보안 정책을 설정할 수 있으며, 이러한 작업을 위하여 모든 프로세스와 파일들에 각각 보안 구문을 첨가하는 메커니즘을 구현한다. 보안 구문은 시스템의 파일이나 디바이스와 같은 오브젝트들에 대한 접근 가능 여부를 정하는 보안 강화 모듈을 리눅스 커널에 포함 시킨다. 시스템 관리자에 의하여 정의된 보안 정책은 보안 서버 프로세스를 통하여 커널에 의해 접근된다. 이 프로세스는 커널의 일부로 실행되며, 어떠한 주체 즉, 어떠한 프로세스 또는 사용자가 시스템의 특정 파일이나 디바이스에 대한 접근이 가능한 지의 여부를 결정한다. 이러한 제어 메커니즘을 보안 강화 리눅스에서는 형태 강화 (TE: Type Enforcement)라고 한다[11]. 한편, 보안 강화 리눅스는 정보의 흐름을 통제하는 다중 보안 모델(MLS: Multi-Level Security Model)을 구현할 수 있다.

3. 제안된 시스템 : 리눅스 보안 모듈을 기반으로 하는 확장된 역할 기반 접근 제어 보안 시스템 (LSM-Based Extended RBAC Security System)

본 논문에서 제안된 시스템의 구조는 다음과 같다.

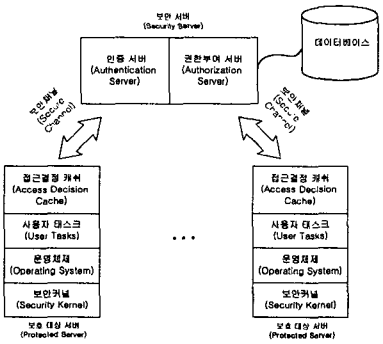

그림 1. 리눅스 보안 모듈 기반의 확장된 역할 기반 접근 제어 시스템

보안 서버는 기능에 따라 인증 서버(Authentication Server)와 권한 부여 서버(Authorization Server)로 나뉘어 진다. 인증 서버는 사용자의 인증을 담당하며 보호 대상 시스템과는 별개의 사용자 정보 테이블과 보호 대상 시스템과의 사용자 매핑을 위한 매핑 테이블을 가지고 있다. 보호 대상 서버로부터 사용자 인증 요청이 들어오면 사용자 매핑 테이블과 정보 테이블을 사용하여 인증 요청 사용자의 정보를 검색한다. 검색 결과는 사용자에 대한 신용 정보(Credential)가 된다. 이 결과를 보호 대상 서버로 보낸다. 보호 대상 서버는 보안 커널을 이식한 서버로 보호되는 대상이 된다. 특정 사용자가 시스템의 자원에 접근하려고 할 때 보안 커널을 통하여 접근 결정이 이루어 진다. 보안 커널(Security Kernel)은 일반 운영 체제의 보안 문제점을 해결하기 위한 모듈로 보안 강화를 지원한다. 보안 채널 (Secure Channel)은 보안 서버로부터 사용자 인증 정보 및 보안 정책 정보가 보호 대상 서버로 전송된다. 안전한 전송을 위하여 보안 채널을 통하여 전송한다. 대표적인 터널링 방식으로 GSSAPI와 SSL 채널 방식이 있다. 접근 결정 캐쉬(Access Decision Cache)는 보안 서버로부터의 인증 정보, 접근 결정 정보를 일정 시간 저장한다. 이는 시스템의 성능적인 측면을 향상시키기 위한 요소로서 항상 보안 서버의 정보와의 동기가 이루어져야 한다. 데이터베이스는 역할 기반 접근 제어 데이터베이스로 기능에 따라 역할 계층 데이터베이스(Role Hierarchy Database), 부분적 다중 계층 보안 데이터베이스(Partial Multi-Level Security Database), 관리 오브젝트 계층 데이터베이스(Management Object Hierarchy Database), 사용자 데이터베이스(User Database), 권한 부여 데이터베이스(Authorization Database), 중요한 자원 및 응용 프로그램 등록 데이터베이스(Important Resource and Application Registration Database)로 나뉘어 진다.

본 고에서 제안된 확장된 역할 기반 접근 제어 시스템의 특징은 다음과 같다. 첫째, 인증의 강화를 위하여 원 타임 패스워드를 사용한다. 한 사용자에 대하여 고정된 패스워드가 설정되는 것이 아니라 수학적 함수의 사용으로 시스템의 파라미터를 통하여 패스워드가 계산된다[6]. 제안된 시스템은 인증과 권한부여 기능을 제공하는 중앙의 보안 서버로부터 시간대별 사용자 인증을 한다. 이 방법은 시스템의 성능적인 문제가 발생할 수 있다. 따라서 그림 2와 같이 중요한 자원 및 응용 프로그램 등록 데이터베이스에 재 인증이 필요한 중요한 자원 및 응용 프로그램을 등록한 후, 이 리스트에 포함된 목록을 기반으로 재 인증을 실행한다.

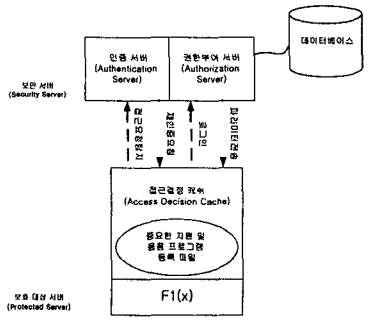

그림 2. 원 타임 패스워드 방식을 이용한 재 인증 방법

둘째, 부분적 다중 계층 보안을 구현하여 역할 계층에서의 정보의 흐름을 통제한다. 본 고에서 제안된 시스템은 일부 중요한 자원에 대하여 정보의 흐름 방향이 높은 역할 계층에서 낮은 역할 계층으로 전송될 수 없도록 설정한다. 이에 대한 정보는 그림 3와 같이 역할 계층 데이터베이스에 다중 계층 보안을 지원하는 부분이 다중 계층 보안 데이터베이스에 설정되어 있으며 이들 역할 계층 데이터베이스 정보와 다중 계층 보안 데이터베이스 정보가 충돌하지 않도록 설정하여야 한다.

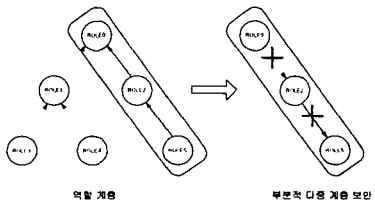

그림 3. 부분적 다중 계층 보안 (Partial Multi-Level Security)

셋째, 확장된 역할 기반 접근 제어 시스템은 임의적 접근 제어를 구현한다. 이 시스템에서 임의적 접근 제어를 구현한다는 것은 역할 기반 접근 제어를 통해서 관리자의 권한을 위임할 수 있다는 것을 의미한다[7]. 이와 같은 권한 위임을 구현하는데 있어서 관리 오브젝트 계층(MOH: Management Object Hierarchy)이 필요하다. 제안된 시스템의 관리 오브젝트 계층은 계층화된 시스템의 파일 시스템 구조와 동일하다. 이 오브젝트 계층을 이용하여 시스템 관리자에 의해 권한이 위임된다. 시스템 관리자에 의해 권한을 위임 받은 특정 사용자는 위임 받은 관리 오브젝트 영역에 대한 관리자가 되며, 또 다른 사용자에게 자신의 권한을 위임할 수 있다. 그림 4는 시스템의 관리 오브젝트 계층을 나타낸다.

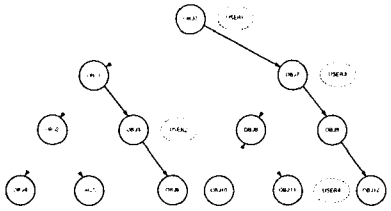

그림 4. 관리 오브젝트 계층 (Management Object Hierarchy)

넷째, 제안된 시스템은 시스템 보안에 대한 감사 정보를 수집하여 설정된 보안 정책의 오류를 판단하고 이에 대응한다. 보안 정책에 오류가 있는 경우 허위 부정 (false negative) 및 허위 긍정 (false positive) 감사 정보의 양이 정상적인 경우의 수보다 월등히 증가하게 된다. 제안된 시스템은 그림 5과 같이 미리 설정한 역할 설정 오류 탐지 정책에 따라 역할을 설정 및 수정한 직후, 특정한 로그가 정상적으로 증가하는 경우 이를 탐지 (Detection)하여 탐지 결과를 시스템의 보안 채널을 통하여 보안 서버로 전송한다. 보안 서버는 각 서버에 대한 오류 탐지 결과들을 분석한 후 적절한 대응 안을 해당 서버로 전송한다. 보안 서버의 응답에 따라 각 보호 대상 서버는 적절한 대응 (Response)을 한 후 해당 로그가 시스템에 영향을 미치지 않도록 삭제 및 백업 등의 조치를 한다. 역할 설정 오류가 탐지된 후, 이에 대한 대응은 정책에 따라 다양하다. 대표적으로 오류 보고서 생성 (Error Reporting), 경고 (Alarm) 생성, 특정 정책 수정 (Policy Modification) 등을 들 수 있다. 특히 특정 정책 수정과 같은 대응 방식은 자동 대응 방식으로 자동 컴퓨팅 (Autonomic Computing) [13]을 사용하여 구현될 수 있다.

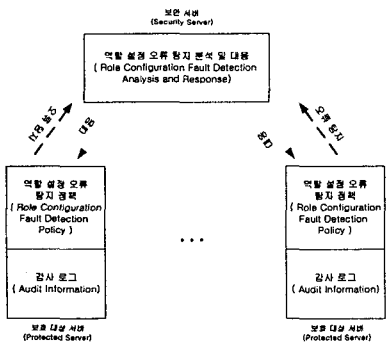

그림 5. 역할 설정 오류 탐지 및 대응 (Role Configuration Fault Detection and Response)

4. 결 론

본 논문에서는 리눅스 보안 모듈 프레임워크를 기반으로 확장된 역할 기반 접근 제어 보안 시스템을 제안하였다. 이 시스템은 보안의 강화를 위하여 원 타임 패스워드 (One-Time Password)의 강화된 인증 방식과 부분적 다중 계층 보안 (Partial Multi-Level Security), 임의적 접근 제어 (Discretionary Access Control) 및 감사 정보를 통한 보안 정책 오류 검사 및 대응 (Security Policy Validation and Response) 기능을 지원한다. 향후 과제로 본 고에서 제안된 보안 모듈을 구현할 예정이다. 특히 마지막 특징으로 제안한 역할 설정 오류 탐지 및 대응의 자동화 연구뿐만 아니라 이를 확장하여 시스템을 보호하는 정책의 디자인 및 적용에 대한 자동화된 방법도 연구하고자 한다.

6. 참고 문헌

[1] D.F. Ferraiolo, J. Cugini, D.R. Kuhn "Role Based Access Control: Features and Motivations" , Computer Security Applications Conference (1995).

[2] D. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli, "A Proposed NIST Standard for Role Based Access Control," ACM Transactions on Information and System Security , vol. 4, no. 3 (August, 2001)

[3] S. Gavrila, J. Barkley, "Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management" (1998), Third ACM Workshop on Role-Based Access Control.

[4] D.R. Kuhn. "Role Based Access Control on MLS Systems Without Kernel Changes" (Kuhn) Third ACM Workshop on Role Based Access Control, October 22-23,1998.

[5] J. Barkley, C. Beznosov, Uppal, "Supporting Relationships in Access Control using Role Based Access Control" , Fourth ACM Workshop on Role-Based Access Control (1999).

[6] Charles P.Pfleeger, Shari Lawrence Pfleeger, Security in Computing, Third Edition, Prentice Hall.

[7] Glenn Faden, " RBAC in UNIX Administraion" , Proceedings of the fourth ACM workshop on Role-based access control October 1999

[8] " RBAC in the SolarisTM Operation Environment" , White Paper, Sun Microsystems, Inc.

[9] C. Wright, C. Cowan, J. Morris, S.Smalley, G.Kroah-Hartman, " Linux Security Module Framework" , http://www.kroah.com/linux/talks/ols_2002_lsm_paper/lsm.pdf

[10] C. Wright, C. Cowan, J. Morris, S.Smalley, G.Kroah-Hartman, " Linux Security Module: General Security Support for the Linux Kernel"

[11] Sys Admin Magazine, http://www.samag.com /documents/s=7835/sam0303a/0303a.htm

[12] A.G.Ganek, T.A.Corbi, The dawning of the autonomic computing era, IBM SYSTEMS JOURNAL, VOL 42, NO 1, 2003.