Journal of Convergence for Information Technology
Vol. 10. No. 10, pp. 32-39, 2020
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2020.10.10.032

클라우드 환경에서 개체 속성 기반 접근제어 모델

최은복
전주대학교 스마트미디어학과 교수

An Entity Attribute-Based Access Control Model in Cloud Environment

Eun-Bok Choi
Professor, Dept. of Smartmedia, Jeonju University

요 약 클라우드 환경의 대규모 인프라 구조에서는 응용프로그램들과 디바이스의 공유로 인하여 불법적인 접근권한 문제가 빈번하게 발생하기 때문에 이러한 공격에 적극적으로 대응하기 위해서는 상황별로 대비가 가능한 강화된 접근통제 시스템이 요구된다. 우리는 대규모 인프라 환경에 기반한 보안등급과 릴레이션 개념의 개체 속성 기반 접근통제 모델을 제시하였다. 본 모델은 주체와 객체에 무결성과 기밀성 등급을 부여하고 동일한 역할에 대해 서로 다른 서비스가 가능한 강화된 접근제어 특성을 가지며, 서비스와 관련된 릴레이션과 상태정보인 컨텍스트에 의해 역할과 권한을 배정함으로써 권한 관리의 유연성을 갖는다. 또한, 대학이라는 대규모 인프라 구조를 갖는 다중 서비스 환경에 적용한 응용 사례를 통하여 본 모델의 적용 가능성을 제시하였다.

주제어 : 클라우드컴퓨팅, 가상화 보안, 접근제어, MAC, DAC, 보안정책, 릴레이션, 컨텍스트

Abstract In the large-scale infrastructure of cloud environment, illegal access rights are frequently caused by sharing applications and devices, so in order to actively respond to such attacks, a strengthened access control system is required to prepare for each situation. We proposed an entity attribute-based access control(EABAC) model based on security level and relation concept. This model has enhanced access control characteristics that give integrity and confidentiality to subjects and objects, and can provide different services to the same role. It has flexibility in authority management by assigning roles and rights to contexts, which are relations and context related to services. In addition, we have shown application cases of this model in multi service environment such as university.

Key Words : Cloud Computing, Virtualization Security, Access Control, MAC, DAC, Security Policy, Relation, Context

1. 서론

데스크톱이나 노트북에서 작동되는 클라이언트와 이들에 서비스를 제공하는 서버로 구축되어 운영되는 네트워크 환경인 클라이언트 서버 컴퓨팅 환경은 가상화 기능을 갖춘 새로운 클라우드 환경으로 서서히 변화해 가는 추세이다. 이러한 클라우드 컴퓨팅 환경은 사용자에게 서비스를 제공하기 위한 서버나 스토리지와 같은 IT 자원을 보유하지 않고 이 같은 자원을 소유한 클라우드 컴퓨팅 플랫폼 제공자를 통해 운영하는 새로운 컴퓨팅 환경을 의미한다[1].

가상화는 응용프로그램의 성능을 최소의 비용으로 최

*Corresponding Author : Eun-Bok Choi(ebchoi@jj.ac.kr)
Received August 24, 2020
Revised September 4, 2020
Accepted October 20, 2020
Published October 28, 2020

적의 효율성을 제공할 수 있는 클라우드 컴퓨팅 환경을 구성하는데 도움이 되는 기술이라 할 수 있지만 응용 프로그램들과 디바이스의 공유로 인한 보안 위험도 제기되고 있는 것이 현실이다. IT 조직의 중요 정보로 구성된 내부 인프라 자원이 클라우드 컴퓨팅 환경으로 이전됨에 따라 가상화 환경에 대한 불법 접근 권한을 획득하기 위한 다양한 우회 활동이 빈번하게 발생하기 때문에 이러한 공격에 대응하기 위해서는 상황별 대처 가능한 새로운 가상화 보안 솔루션이 개발되어야 한다. 특히, 수많은 사용자들의 원격 접속으로 구성되는 클라우드 서비스 환경에서는 이용하는 단말 및 응용프로그램의 다양성과 클라우드에 탑재된 정보의 공유와 위임 등으로 인한 정보 보안의 위협요소가 존재하므로 안전한 클라우드 환경을 제공하기 위해서는 해당 정보의 비밀성, 무결성, 가용성을 제공하는 접근제어기법이 필요하다[2,3].

비즈니스 영역이 급속도로 성장하고 사용자와 다양한 자원에 대한 다중 연결성이 커지는 대규모 인프라 구조 환경에서 접근통제 시스템에 대한 확장성이 절실하다고 본다. 우리는 대규모 인프라 환경에 기반한 일반적인 서비스 개념과 다양한 개체를 갖는 개체 속성 기반 접근통제 모델을 제시하고 대학이라는 대규모 인프라 구조를 갖는 다중 서비스 환경을 본 모델에 적용한 응용 사례를 제시한다.

인가된 연산을 명확히 정의하기 위해 주체와 객체에 대한 무결성 등급과 기밀성 등급을 정의함으로써, 동일한 역할에 대해 서로 다른 서비스가 가능한 강화된 접근제어 모델을 제시한다. 또한, 각 부서들은 서비스와 속성들의 집합인 연산으로 구성되며 하나 이상의 서비스와 연관되어 유기적인 연관성을 갖는데, 특정 조직이나 부서에서 보안이나 통제 정책을 준수하는지 여부를 판단하기 위해 릴레이션을 통해 해당 당사자의 상태정보를 통하여 해당서비스의 수행 여부를 검증하므로써 상위관리의 부담을 덜어주는 유연성을 제공한다.

2. 관련연구

2.1 클라우드 컴퓨팅 보안

클라우드 컴퓨팅은 자원 공유에 중점을 둔 네트워크 기반 환경으로, 서비스로 제공되는 애플리케이션과 해당 서비스를 제공하는 인터넷에 기반한 데이터 센터의 하드웨어 및 소프트웨어를 의미하며, 비즈니스 어플리케이션이 클라우드로 전환되기 위해서는 가상화 기술의 자원 공유 기능과 결합된 서비스를 통해 실현될 뿐만 아니라, 성능향상을 위해서 사용자 인증을 포함한 접근제어, 시스템 운용시 발생되는 취약점 분석과 패치 구성 그리고 사고에 대한 대응과 응답 등 같은 기술과 관련된 관리 기능이 충족되어야 한다[4].

- •접근 제어
- •취약점 및 문제
- •패치 및 구성
- •사고대응책 및 적정 응답
- •클라우드 시스템 사용 및 접근 모니터링

클라우드 컴퓨팅 기술을 제공하는 클라우드 공급업체의 경우 보안, 개인의 사생활 보호, 익명성, 통신 용량 및 부하, 신뢰성 확보 등과 같은 중요한 여러 정책이 있을 수 있는데, 이중 가장 선결되어야 하는 것은 보안이라 할 수 있다. 클라우드 컴퓨팅에는 일반 사용자, 학계 및 기업과 같은 여러 고객이 있을 수 있는데, 특히 학계나 연구와 관련된 고객일 경우 클라우드 공급업체는 보안 기능과 더불어 성능향상 방안도 함께 제공될 수 있는 방법을 찾아야 한다[5].

2.2 클라우드 컴퓨팅 접근제어

네트워크에 기반한 클라우드 컴퓨팅은 기존의 인터넷상에 존재하는 보안 문제뿐만 아니라, 다양한 사용자들이 물리적 자원, 응용소프트웨어, 디바이스 등을 서로 공유하는 클라우드 기반의 보안 문제가 존재하기 때문에 이를 해결하기 위한 개선되고 변경되어야 할 특성들이 존재하게 된다[6].

이러한 클라우드 환경의 성능향상을 위한 관리 기능 중 하나인 접근제어 메커니즘은 인증된 사용자에게는 정보 시스템에 정당한 액세스가 가능하게 하고 부당한 사용자로부터의 시스템 무단 액세스를 차단하는 기능으로서, 정보 시스템 및 서비스에 대한 접근 권한 할당을 통제하기 위한 공식적인 절차기법이 기술된다. 접근제어 메커니즘은 신규 사용자의 초기 등록에서 더이상 정보 시스템 및 서비스에 대한 액세스가 필요하지 않은 사용자의 최종 등록 해제에 이르기까지 사용자 액세스 수명주기의 모든 단계를 포함해야 한다. 특별히, 특정한 사용자에 의해 시스템 제어권이 번복되거나 영향을 미치는 특정 접근 권한에 대한 적정한 권한 조정이 제어

될 필요성이 있다[7].

클라우드 컴퓨팅을 제공하는 기업에서는 가상자원을 안전하게 제공하고 관리하기 위해 접근제어 메커니즘이 적용되는데, 아마존의 EC2[8] 제품은 안전하고 규모 조정이 가능한 클라우드 기반 웹서비스로서, ID와 상황 기반 접근제어 기법을 통해 개발자가 더 쉽게 웹 규모의 클라우드 컴퓨팅 작업을 할 수 있도록 설계되었으며, S3[9]는 확장성과 데이터 가용성 및 보안과 성능을 제공하는 객체 스토리지 서비스로, 객체 등급 접근제어 기법으로 사용자에게 자신의 모든 데이터에 대한 접근 권한을 제어할 수 있는 접근제어 리스트를 통하여 허가하는 정책을 사용한다. VMWare의 주력제품인 VirtualCenter[10]은 주어진 작업이 사용자의 직무에 의해 정해지는 역할기반 접근제어기법에 기반한 작업 기반 권한 관리 시스템을 제공한다. 클라우드 컴퓨팅 환경에서 직무에 기반한 역할기반 접근제어 기법이 적용되기 위해서는 사용자 계정 권한이 적법하게 설정되고 역할과 책임이 동반되는 직무에 기반한 최소 권한 규칙이 이행되어야 한다[11].

2.3 강제적 접근제어

접근제어란 인증된 개체에는 인가된 해당 릴레이션을 수행하게 하고, 허가되지 않은 릴레이션은 수행하지 못하도록 하는 메커니즘을 의미하고, 접근제어 모델은 보안 정책과 구현을 연결하는 인터페이스로서, 정의된 정책을 허락하는 프로세스라고 볼 수 있다. 다양한 정보의 연결과 공유를 위한 사용자들의 요구 증대로 인해 거의 모든 기업들에 의해 제공되는 온라인 서비스들은 서비스에서 서비스로 확장되어가면서 하루가 다르게 폭증하고 있다. 따라서 대규모 서비스 인프라 환경에서는 강력하고 효율적이며 안전한 접근제어 및 데이터 공유 시스템을 갖추어야 한다[12].

접근제어는 주체와 객체의 접근방법을 기술하고 있는 확정된 보안 정책의 목적과 규칙을 이행하기 위한 모델로서 임의적 접근제어(DAC : Discretionary Access Control)와 역할기반 접근제어(RBAC : Role-Based Access Control), 강제적 접근제어(MAC : Mandatory Access Control) 등이 있다.

임의적 접근제어모델은 주체가 자신이 관리대상인 객체나 자원에 대한 작업을 제어할 수 있도록 하는 정책으로 일반적으로 권한이 부여된 접근제어리스트(ACL:Access Control List)를 통해 통제된다[13]. 역할기반 접근제어모델은 비임의적 모델이라고도 하는데, 관리자는 사용자를 위한 컨테이너나 그룹핑 역할을 수행하는 역할에 할당된 권한에 따라 접근 여부가 결정되는 모델이다[14]. 강제적 접근제어모델은 데이터 소유자에게 접근 결정을 부여하는 대신, 접근 부여와 거부에 대한 권한을 주체의 인가등급과 객체의 보안등급을 시스템이 비교 판단하게 함으로써 보안성을 강화한 모델이라 할 수 있다. 강제적 접근제어모델의 구성요소는 Fig.1과 같이 접근집행함수, 참조모니터 그리고 접근제어정책으로 구성된다. 주체의 자원에 대한 접근 요청이 발생하면 접근제어함수가 호출되어 주체의 자원 요청이 인터셉트되고 인터셉트된 정보를 참조모니터로 전달하게 된다[13].

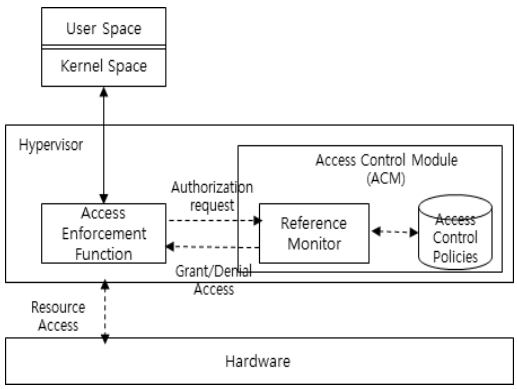

Fig. 1. Architecture of Mandatory Access Control

모든 자원의 접근 요청에 대한 모니터링 기능을 하는 참조 모니터는 주체와 자원에 해당하는 객체의 접근 권한이 관리자에 의해 정의된 규칙 집합으로 구성된 접근제어 정책에 의해 접근제어요청에 대한 접근 여부를 결정하게 된다[15,16].

3. 개체속성기반 접근제어

본 개체속성기반 접근제어 모델의 컴퓨팅 환경을 구성하는 개체로는 가상머신, 주체, 객체 그리고 이들과의 관계를 형성하는 역할과 서비스 그리고 권한 등이 있다. 주체는 보안 모델 관점에서 능동 개체이며 역할은 조직이나 조직원들이 수행하는 기능을 말한다. 서비스는 연산과 연관되는 릴레이션의 집합이며, 권한은 모델을 구체화하기 위해 객체인 구성원이나 사용자에 영향

을 받지 않는 그룹핑된 역할들이라 볼 수 있다.

3.1 개체속성기반 접근제어 정의

주체, 객체, 자원 등을 포함하는 개체들의 속성이 정형적으로 구성되어야 비로소 해당 속성들의 권한과 상태정보, 보안등급에 따라 객체의 접근 여부가 안전하게 필터링 될 수 있으므로 다음과 같이 개체들의 속성을 정의한다.

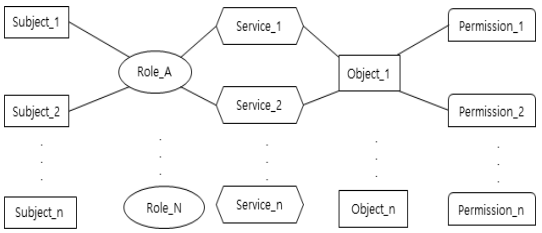

Fig. 2. Authorization for the Entity Based Access Control

S = { 주체 또는 사용자들의 집합 }
O = { 객체 또는 자원들의 집합 }
M = { 주체에 대한 객체의 접근 모드 }, S × O → { ∅. {read}, {write}, {read,write}}
L = {기밀성 보안등급, 무결성 보안등급}, C_level → { U≤C≤S≤TS }, I_level → { C≤I≤VI }
Re = { 릴레이션들의 집합 }, {Environ, Sub_Grant, Obj_Grant, Restrict, Compare, Permission}

일반적으로 역할기반접근제어모델인 RBAC 시스템은 임무분리를 통해 상호 배타적인 역할을 수행토록 구성된다. 즉, 하나의 역할에 서로 다른 주체들이 배정되는 경우 해당 주체들의 이익충돌이 발생하게 되면 서로 다른 역할로 구분하거나 선결조건이나 동적임무분리 정책에 의거하여 세션 수행을 분리한다. 본 모델에서는 이와같이 하나의 역할에 서로 다른 주체들이 배정되는 경우 서비스를 통해 상황에 적합한 권한 배정이 가능해야 한다. 만약 조직이 2명의 서로 다른 주체 subject_1과 subject_2에게 동일한 하나의 역할 Role_A에 대한 권한을 부여한다고 가정하자. 본 모델에서는 서로 다른 2개의 서비스 배정을 통해, Fig.2과 같이 (Role_A, Sservice_1)은 subject_1에게 사용 권한 Permission_1을 부여하고, (Role_A, Service_2)은 subject_2에게 권한 Permission_2를 부여함으로써 다양하고 세밀한 권한 표현이 가능하다

우리는 상이한 주체에게 동일한 역할을 배정함으로 인한 주체의 기밀성과 객체의 무결성 보장의 문제를 해결하기 위해 BLP 및 Biba 흐름제어 모델을 사용할 것이며, 권한 배정은 기밀성과 무결성에 대한 등급 수준에 따라 주체에게 할당될 것이다. 이러한 연관성을 정의하기 위해, 우리는 권한 관계에서 주체의 무결성과 관련한 등급 I-level과 기밀성과 관련된 등급 C_level을 정의한다.

기밀성 보안정책은 데이터 비밀 보호를 위한 참조 정책으로 각 등급은 두가지 구성요소에 의해 정해지는데, 하나는 기밀성 등급이고 또다른 하나는 범주의 집합으로 기밀성 등급은 TS〉S〉C〉U의 관계로 구성된다. 여기에서 U는 Unclassfied, C는 Confidential, S는 Security, TS는 Top Security를 의미하며, 범주의 집합은 개체들의 비계층적인 구조를 갖는 부분집합으로 조직의 구성 환경에 의해 형성되어 진다. 무결성 보안정책은 데이터 임의변경방지를 위한 정책으로 무결성 보안등급을 Crucial(C), Very Important (VI), Important(I)로 구분하며 C〉VI〉I의 관계를 형성한다. 범주의 집합은 BLP모델과 마찬가지로 비계층적인 구조 관계를 갖는다.

기밀성과 무결성을 대변하는 BLP 모델과 Biba 모델에 기반하여 물리적 자원이나 디지털 자원을 대표하는 개체와 연관된 권한 연산을 정의하기 위해서는 새로운 조직의 추가, 수정 그리고 소멸시 무결성과 기밀성에 기반한 개체를 재배정하고 이와 연관된 자원들을 재분류해야 정확하고 일관된 자원 접근이 제공될 것이다. 조직은 연산이나 릴레이션을 통해 세부적으로 자원을 분류함으로써, 주체의 무결성과 더불어 자원의 기밀성을 확보할 수 있다.

본 모델에서는 아래와 같이 서비스 개념과 연관된 릴레이션과 기밀성과 무결성을 판단하는 알고리즘을 정의한다.

•Environ 릴레이션

가상머신 환경에서 역할은 정책과 관련된 특성을 수행하는 주체들과 연관되며, 퍼미션은 릴레이션들의 집합으로써 동일한 보안규칙이 적용되어야 하며 아래와 같은 가상머신에 전반적으로 적용이 가능한 모든 연산을 Environ 릴레이션으로 정의한다.

Environ(VMn, Re, Pe, Se) → 가상머신(VM)은 하나 이상의 서비스(Se) 프레임워크 내에서 퍼미션(Pe)의 일부분인 릴레이션(Re)으로 간주된다. 각 부서들은 서비스와 릴레이션들의 집합인 퍼미션으로 구성되며 하나 이상의 서비스와 연관되어 유기적인 연관성을 갖는다. 예를 들어 특정 가상머신의 경우 인사부서의 사원개인정보서비스(personnel)의 인적조회릴레이션(human_resource)은 read, write 퍼미션으로 구성된다고 가정할 경우 Environ(VM, human_resource, read/write, personnel) 릴레이션으로 구성된다.

•Sub_Grant 릴레이션

Sub_Grant(VMn, s, r, Se, level) → 가상머신은 주체에게 특정 서비스에 대한 등급과 역할을 수행할 수 있는 권한을 부여한다. 여기서 Se는 서비스, VMn은 가상머신, s는 주체, level은 기밀성과 무결성 등급 그리고 r은 역할을 의미하며, 가상머신이 어떤 주체에게 특정 서비스에 대한 역할을 수행하도록 허용한다는 것으로 해석할 수 있다. 따라서, 다양한 서비스를 가진 가상머신 내에서 그 조직원의 기능이나 지위나 상태는 수행하고자 하는 업무와 관련하여 변경될 수 있다.

•Obj_Grant 릴레이션

인가된 연산을 명확히 정의하기 위해 객체에 대한 무결성과 기밀성에 대한 level 매개변수를 추가함으로써, 동일한 객체의 뷰 연산에 대해 접근통제 형태가 상이한 두 가지 릴레이션을 정의한다. 예를 들어 무결성과 기밀성에 따른 뷰 권한의 경우, Obj_Grant(VMn, o, v, Se, I_level) → 가상머신(VMn)은 서비스(Se)에서 무결성 등급에 기반한 객체(o)의 뷰(v) 권한을 사용한다.

Obj_Grant(VMn o, v, Se, C_level) → 가상머신(VMn)은 서비스(Se)에서 기밀성 등급에 기반한 객체(o)의 뷰(v) 권한을 사용한다.

•Restrict 릴레이션

권한 부여에 대한 연산 수행은 위치정보에 따른 선결 조건 그리고 특정 기간이나 일부 시간에 따른 제약조건과 같이 세부 조건이나 상황 컨텍스트 정보에 따라 영향을 받는다. 예를 들어, Restrict(VMn, s, Re, o, c, Se) → 가상머신은 주체(s)가 서비스(Se)에서 객체(o)에 대한 릴레이션(Re)를 특정한 컨텍스트(c)에 한해 제한 적용한다.

•Compare 릴레이션

Compare 릴레이션은 주체와 객체에 대한 기밀성과 무결성 등급을 비교판단한다. 예를 들어, Compare(s, I_level ≤ o, I_level)은 주체의 무결성 등급과 객체의 무결성 등급을, Compare(s, C_level ≤ o, C_level)은 주체의 기밀성 등급과 객체의 기밀성 등급을 비교판단한다.

•Permission 릴레이션

Permission은 제한적인 컨텍스트 상황에서 특정 서비스의 객체에 대한 릴레이션을 수행할 수 있는 허가사항으로 특정 요구나 수요에 반응하는 특정 상황과 일반 상황에 대한 허가와 금지로 구분한다. 예를 들어, Permission(VMn, r, Re, ac, v, c, Se) → 가상머신은 서비스(Se)에 대해 컨텍스트(c) 특정 상황으로 뷰(v)에서 퍼미션(ac)의 일부분인 릴레이션(Re)를 수행할 수 있는 역할(r) 권한을 부여한다. 또한, Permission(VMn, r, Pe, v, c, Se) → 가상머신은 서비스(Se)에 대해 컨텍스트(c) 일반 상황으로 뷰(v)에서 전반적인 퍼미션(Pe)을 수행할 수 있는 역할(r) 권한을 부여한다.

Table 1. Confidentiality Assurance Algorithm

```
C_Level = with U | C | S |TS ;
Access_mode = with ∅, read, write;
Var access: Access_mode;
if
Environ(VMn, Re, Pe, Se),
Sub_Grant(VMn, s, r, Se, C_level),
Obj_Grant(VMn, o, v, Se, C_level),
Restrict(VMn, s, Re, o, c, Se),
Permission(VMn, r, Re, ac, v, c, Se),
Compare(s, C_level ≥ o, C_level) then
access = read;
else if Compare(s, C_level ≤ o, C_level)
access = write;
else
access = ∅;
```

기밀성 검증 알고리즘(Table 1)의 등급은 U | C | S | TS로 구성되며 접근 모드는 연산 모드를 갖지 않는 공백과 read, write 연산으로 구성된다. 만약, 특정 가상머신의 연산 권한 지정 여부에 대해서는 먼저, Environ 릴레이션부터 Permission 릴레이션까지 각 릴레이션에 대한 설정 여부를 판단한 다음, 주체의 기밀성 등급이 객체의 기밀성 등급을 지배할 경우 read 연산을, 주체의 기밀성 등급이 객체의 기밀성 등급에

지배될 경우 write 연산을, 그 이외에는 공백모드를 부여하게 된다.

Table 2. Integrity Assurance Algorithm

```
I_Level = with C | VI | I ;
Access_mode = with ∅, read, write;
Var access: Access_mode;
if
Environ(VMn, Re, Pe, Se),
Sub_Grant(VMn, s, r, Se, I_level),
Obj_Grant(VMn, o, v, Se, I_level),
Restrict(VMn, s, Re, o, c, Se),
Permission(VMn, r, Re, ac, v, c, Se),
Compare(s, I_level ≤ o, I_level) then
access = read;
else if Compare(s, I_level ≥ o, I_level)
access = write;
else
access = ∅;
```

무결성 검증 알고리즘(Table 2)의 등급은 C | VI | I로 구성되며, 특정 가상머신의 연산 권한 지정 여부에 대해서는 기밀성 알고리즘과 같이 각 릴레이션에 대한 설정 여부를 판단한 다음, 주체의 무결성 등급이 객체의 무결성 등급에 지배되는 경우 read 연산을, 주체의 무결성 등급이 객체의 무결성 등급을 지배할 경우 write 연산을, 그 이외에는 공백모드를 부여하게 된다.

3.2 개체속성기반 접근제어 응용

학생과 교직원에게 다양한 서비스를 제공하는 대학과 같은 대규모 조직을 구성하는 환경에서 본 모델에서 적용한 몇가지 릴레이션을 알아보자. 만약 특정 등급의 주체(Staff)가 교무처(Dep_Academic) 산하 교학과(Education)의 학적업무서비스(School_service)의 학적부 파일(School_register)에 접근 모드인 read와 write 연산을 실행하기 위해서는 다음 관계가 반드시 만족되어야 한다.

▷ Environ(Univ, School_service, Education, Dep_Academic) : 대학(Univ)은 교무처(Dep_Academic)의 학적업무서비스(School_service)가 교학과(Education)에 배정되었는지를 확인하고,

▷ Sub_Grant(Univ, Staff, Education, Dep_Academic, Level) : 대학(Univ)은 교학과(Education)의 Staff에게 특정 등급이 부여되었는지를 확인하고,

▷ Obj_Grant(Univ, School_register, schooling, Level) : 대학(Univ)은 객체인 학적부 파일(School_register)의 특정 등급이 부여되었는지를 확인하고,

▷ Restrict(Univ, Staff, read/write, School_register, prerequisite_constraint, Education) : 대학은 Staff와 School_register 간의 선결 조건을 점검하고, Compare(Staff, C_level ≥ School_register, C_level) AND Staff, I_level ≤ School_register, I_level) : read 연산의 경우 Staff의 기밀성 등급이 School_register의 기밀성 등급을 지배하고, Staff의 무결성 등급이 School_register의 무결성 등급에 지배되는 경우, Compare(Staff, C_level ≤ School_register, C_level) AND Staff, I_level ≥ School_register, I_level) : write 연산의 경우 Staff의 기밀성 등급이 School_register의 기밀성 등급에 지배되고, Staff의 무결성 등급이 School_register의 무결성 등급을 지배하는 경우, Permission(Univ, Education, read, write, School_service, prerequisite_constraint, temporary_ constraint) : 교학과(Education)의 학적업무서비스(School_service)의 학적부 파일(School_register)에 대한 연산이 허용된다.

다음 Table 3에서는 제시된 대학 조직의 환경을 본 개체속성기반 접근제어모델인 EABAC(Entity Attribute- Based Access Control)에 적용할 경우의 권한 릴레이션을 기술하였으며, 이는 무결성 및 기밀성 등급과 역할 및 서비스로 구성된 릴레이션에 의해 상위 관리의 부담을 덜어주는 유연성과 권한관리의 편의성을 제공하는 강화된 접근제어 보안 모델을 갖는다.

Table 3. Permission in the EABAC

Environ(Univ, School_service, Education, Dep_Academic), Sub_Grant(Univ, Staff, Education, Dep_Academic, level), Obj_Grant(Univ, School_register, schooling, level), Restrict(Univ, Staff, read/write, School_register, prerequisite_constraint, Education), Compare(Staff, C_level ≥ School_register, C_level) AND Staff, I_level ≤ School_register, I_level) → Permission(Univ, Education, read, School_service, prerequisite_constraint, temporary_constraint)
Environ(Univ, School_service, Education, Dep_Academic), Sub_Grant(Univ, Staff, Education, Dep_Academic, level), Obj_Grant(Univ, School_register, schooling, level), Restrict(Univ, Staff, read/write, School_register, prerequisite_constraint, Education), Compare(Staff, C_level ≤ School_register, C_level) AND Staff, I_level ≥ School_register, I_level)→ Permission(Univ, Education, write, School_service, prerequisite_constraint, temporary_constraint)

4. 결론

사용자와 다양한 자원에 대한 다중 연결성이 커지는 클라우드 환경의 대규모 인프라 구조에서 응용프로그

램들과 디바이스 공유에 대한 효율성과 더불어 보안 위험도 제기되고 있는 것이 현실이다. 특히 가상화 환경에 대한 불법적인 접근 권한 획득에 의한 다양한 우회 활동이 빈번하게 발생하기 때문에 이러한 공격에 대응하기 위해서는 상황별 대처 가능한 새로운 접근통제 시스템에 대한 확장성이 절실하다.

우리는 대규모 인프라 환경에 기반한 일반적인 서비스 개념과 다양한 개체를 갖는 개체 속성 기반 접근통제 모델을 제시하고 대학이라는 대규모 인프라 구조를 갖는 다중 서비스 환경을 본 모델에 대한 적용한 응용 사례를 제시한다.

인가된 연산을 명확히 정의하기 위해 주체와 객체에 대한 무결성 등급과 기밀성 등급을 정의함으로써, 동일한 역할에 대해 서로다른 서비스가 가능한 강화된 접근제어 보안 모델을 제시하였다. 각 부서들은 서비스와 속성들의 집합인 연산으로 구성되며 하나 이상의 서비스와 연관되어 유기적인 연관성을 갖는데, 특정 조직이나 부서에서 보안이나 통제 정책을 준수하는지 여부를 판단하기 위해 릴레이션과 해당 서비스의 상태정보를 통하여 서비스의 수행 여부를 검증함으로 상위관리의 부담을 덜어주는 유연성을 제공한다. 또한, 역할과 직접적으로 연관되는 권한설정보다는 컨텍스트와 서비스를 통한 역할과 권한설정을 통하여 새로운 권한을 추가하거나 다른 권한을 삭제하는 등 권한 관리의 편리성과 유연성을 제공할 수 있을 뿐만 아니라 다양한 접근통제 기법으로 넓힐 수 있는 확장성을 갖는다.

REFERENCES

[1] R. Aluvalu & L. Muddana. (2016). A Dynamic atttribute-based risk aware access control model(DA-RAAC) for cloud computing, *IEEE International Conference on Computational Intelligence and Computing Research(ICCIC),* DOI : 10.1109/iccic.2016.7919618

[2] G. Sala, D. Sgandurra & F. Baiardi. (2007). Security and Integrity of a Distributed File Storage in a Vitual Environment, *Fourth Inernational IEEE Security In Storage Workshop,* 58-69. DOI : 10.1109/SISW.2007.10

[3] G. Cheng, H. Jin, D. Zou, A. Ohoussou & F. Zhao. (2008). A Prioritized Chinese Wall Model for Mananging the Covert Information Flows in Virtual Machine Systems, *The 9th Inernational Conference for Young Computer Scienttists*, 1481-1487. DOI : 10.1109/ICYCS.2008.534.

[4] F. Sabdhi. (2011). Cloud Computing Security Threates and Responses,, I*nernational Conference on Communication Software and Networks(ICCSN)*, 245-249.

[5] F. Sabdhi. (2011). Virtualization-Lever Security in Cloud Computing, *Inernational Conference on Communication Software and Networks(ICCSN),* 250-254.

[6] T. Shinagawa, H. Eiraku, S. Hasegawa, K. Omote, K. Tanimoto, T. Horie & K. Kato. (2008). Introducing Role-based Access Control to a Secure Virtual Machine Monitor :Security Policy Enforcement Mechanism for Distributed Computers, *2008 IEEE Asia-Pacific Services Computing Conference,* 1225-1230. DOI : 10.1109/APSCC.2008.14

[7] C. Musca, A. Ion, C.Leordeanu & V. Cristea. (2013). Secure Aceess to Cloud Resource RBAC in Cloud System, *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing,* 554-558.

[8] http://aws.amazon.com/ec2.

[9] http://aws.amazon.com/s3.

[10] http://aws.vmware.com/products/vcenter-server.html

[11] British Standards. (2013). *ISO/IEC 27001: 2013: Information Technology—Security Techniques—Information Security Management Systems--Requirements.* International Organization for Standardization.

[12] C. Pengrui, W. LingDa, Y. Chao & Y. Ronghuan. (2016). A Hierachical Access Control Model of Software Repository Based on RBAC, *IEEE*, 761-765. DOI : 10.1109/icsess.2016.7883179

[13] T. Win, H Tianfield & Q. Mair. (2014). Virtualization Security Combining Mandatory Access Control and Virtual Machine Introspection, *2014 IEEE/ACM 7th International Conference on Utitiy and Cloud Computing,* 1004-1009.

[14] Y. Sanches, S. Demurjian & M. Baihan. (2019). A Service-based RBAC & MAC approach incorporate into the FHIR standard, *Digital Communications and Networks, 5*, 214-225. DOI : 10.1016/j.dcan.2019.10.004

[15] E. Choi & S. Lee(2016), Access Control

Mechanism based on MAC for Cloud Convergence, *Journal of the Korea Convergence Society, 7(1),* 1-8.
DOI : 10.15207/jkcs.2016.7.1.001

[16] B. Taubmann, N. Rakotondravony & H. Reiser, (2016), CloudPhylactor:Harnessing Mandatory Access Control for Virtual Machine Introspection in Cloud Data Centers, *2016 IEEE TrustCom-BigDataSE-ISPA*, 957-964.
DOI : 10.1109/TrustCom.2016.160

[17] E. Choi, (2018), A Virtualization Management Convergence Access Control Model for Cloud Computing Environments, *Journal of Convergence for Information Technology, 8(5),* 69-75.
DOI : 10.22156/CS4SMB.2018.8.5.069

[18] X. Ding & J. Yang, (2019), An Access Control Model and Its Application in Blockchain, *2019 International Conference on Communications, Information System and Computer Engineering (CISCE),* 163-167.
DOI : 10.1109/CISCE.2019.00044

최 은 복(Eun-Bok Choi) [정회원]

· 1992년 2월 : 전남대학교 전산학과 (이학사)
· 1996년 2월 : 전남대학교 전산학과 (이학석사)
· 2000년 8월 : 전남대학교 전산학과 (이학박사)
· 2002. 2월 ~ 현재 : 전주대학교 스마트미디어학과 교수

· 관심분야 : 통신망관리, 가상화보안, 접근통제
· E-Mail : ebchoi@jj.ac.kr