KNOM Review '16-02 Vol.19 No.02
https://doi.org/10.22670/knom.2016.19.2.36

서비스 매쉬업 시스템을 위한 동적 생성 역할 기반 사용자 접근제어 모델 설계

임 이 삭◆, 권 동 우*, 안 동 혁*, 주 홍 택°

Design of a User Access Control Model Based on Dynamically Created Roles for a Service Mashup System

Isaac Lim◆, Dongwoo Kwon*, Donghyeok An*, Hongtaek Ju°

요 약

서비스 매쉬업 시스템은 기존의 단일 서비스들이 제공하는 Open API 들을 서로 연결하여 새로운 매쉬업 서비스를 생성하고 제공하는 시스템이다. 매쉬업 서비스는 기존의 단일 서비스의 접근제어와 달리 계층적으로 접근제어 연결이 이루어져 있다. 접근제어를 위한 모델 중 임의적 접근제어(discretionary access control, DAC)와 강제적 접근제어(mandatory access control, MAC)는 서비스 매쉬업 시스템과 같이 잦은 권한 변경 요청이 필요하고 규모가 큰 시스템에서의 접근제어 모델링에는 비효율적이다. 또 다른 접근제어 모델인 역할 기반 접근제어(role-based access control, RBAC)은 시스템 자원과 사용자의 요구가 증가함에 따른 역할의 복잡성 증가에 대하여 문제점이 발생한다. 이러한 문제점을 해결하기 위해 본 논문에서는 동적으로 생성되는 역할을 기반으로 한 동적역할 기반 접근제어 모델(dynamic role-based access control, DRBAC)을 제안한다. 제안하는 모델에서 매쉬업 서비스 개발자는 새롭게 정의되는 매쉬업 서비스에 포함될 Open API 서비스들을 선택하고 매쉬업 서비스의 이름을 등록한다. 이에 따라, 역할이 동적으로 생성되고 개발자에게 할당되어 해당 매쉬업 서비스를 개발하는데 필요한 Open API들의 사용 권한을 얻을 수 있으며, 서비스 매쉬업 시스템에서의 역할 관리 복잡성 문제를 해결할 수 있다.

Key Words : RBAC, Service Mashup System, DRBAC, Dynamic Role, User Access Control

ABSTRACT

A service mashup system generates and provides a new mashup service, connecting open APIs that existing services provide. The mashup service is hierarchical access control connection, unlike the existing single service access control. When privilege changes are frequent in service mashup systems, using discretionary access control (DAC) and mandatory access control (MAC) is very inefficient. Another access control model, role-based access control (RBAC), manages user access control through roles, which compensates for the disadvantages of the existing access control models. However, the problems arise in terms of increasing complexity of roles as system resources and user demands increase. To solve these problems, this paper proposes dynamic role-based access control (DRBAC). A service developer selects required open APIs and registers a mashup service. The new role is dynamically created and assign it to the developer. The developer can use the open APIs to develop the mashup service. As a result, DRBAC based on dynamic roles reduces the complexity of the role managements in a service mashup system.

※이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(R0126-16-1009, ICBMS 플랫폼 간 정보 모델 연동 및 서비스 매쉬업을 위한 스마트 중재 기술 개발).
◆ First Author : Keimyung University, Department of Computer Engineering, islim2254@kmu.ac.kr
° Corresponding Author : Keimyung University, Department of Computer Engineering, juht@kmu.ac.kr
* Keimyung University, Department of Computer Engineering, {dwkwon, donghyeokan}@kmu.ac.kr
논문번호 : KNOM2016-02-05, Received September 24, 2016; Revised October 10, 2016; Accepted November 14, 2016

Ⅰ. 서 론

오늘날 네트워크의 발전과 함께 사용자들의 요구를 충족하기 위하여 다양한 서비스들이 개발되고 있다. 하지만 늘어나는 사용자들의 요구사항을 제한된 환경에서 충족시키기 힘들어지자 개발자들은 효율적으로 서비스를 개발하는 방법을 찾게 되었다. 효율적으로 서비스를 개발하는 방법 중 하나인 매쉬업[1]은 기존의 단일 서비스에서 제공하는 Open API들을 조합하여 새로운 서비스를 개발하는 방법이다.

그림 1은 매쉬업을 기반으로 한 서비스 매쉬업 시스템[2,3]으로써 사물 인터넷, 클라우드, 빅데이터, 모바일 그리고 보안 서비스의 플랫폼을 기반으로 하여 다양한 서비스들을 제공한다[4]. 서비스 매쉬업 시스템은 단순히 기존의 단일 서비스만의 결합이 아닌 시스템에서 제공하는 다양한 서비스들의 Open API를 사용하여 개발의 어려움과 개발 비용을 줄여 매쉬업 서비스를 개발할 수 있도록 도와준다. 서비스 매쉬업 시스템은 기존의 단일 서비스의 접근제어와 달리 계층적으로 접근제어 연결이 이루어지며 매쉬업 서비스를 구성하는 모든 단일 서비스의 접근제어를 구성한다.

접근제어란 시스템에서 요구하는 보안을 관리하고[5,6] 사용자의 접근통제 등 다양한 방면에서 사용된다. 사용자 접근제어 방법에는 다양한 모델들이 존재하고 각각의 접근제어 모델들은 서로 다른 특징을 가진다[7]. 서비스 매쉬업 시스템에 적용하기 위한 접근제어 방법 중 강제적 접근제어(mandatory access control, MAC)와 임의적 접근제어(discretionary access control, DAC)[8,9]에서 제공되는 방법은 사용자의 다양한 요구와 시스템에서 제공되는 서비스들을 사용하여 매쉬업 서비스를 개발하기 위한 시스템에는 적합하지 않다.

예를 들어 강제적 접근제어는 주로 엄격한 접근제어를 필요로 하는 곳에서 주로 사용된다. 시스템 주체인 사용자에게 등급을 부여하고 객체인 접근 가능한 시설들에게 적절한 보안등급을 부여한 뒤 설정된 주체와 객체의 등급을 비교해 접근 권한을 부여하는 방식을 가진다. 강제적 접근제어를 서비스 매쉬업 시스템에 적용하게 되면 서로 다른 사용자가 매쉬업 서비스를 개발하기 위해 서비스 개발자라는 같은 등급을 가지지만 서로가 개발을 위해 기능에 따라 필요로 한 서비스의 종류가 다른 상황이 발생 할 수 있다[10]. 이러한 상황을 피하기 위해서는 서로간의 등급을 비교 하는 것 외에 주체와 객체의 상세한 관계 설정이 필요하다. 하지만 강제적 접근제어의 특성상 서비스 매쉬업 시스템에서 접근제어의 세밀한 모델링은 불가능하다. 또한, 임의적 접근제어는 하나의 권한에 대하여 여러 사용자가 필요로 할 때 주로 사용되며 각 사용자에게 객체의 접근 권한을 기술하는 방법으로서 사용자는 자신에게 허용된 권한을 통해 객체에 접근이 가능한 장점이 있다. 이러한 장점은 서비스 매쉬업 시스템에서도 필요하지만 사용자가 매쉬업 서비스의 개발, 시스템 관리, 시스템 개발과 같이 자신이 필요한 권한 외에 불필요한 권한을 갖게 되는 상황을 갖게 될 수 있으며, 하나의 서비스 뿐 아닌 두 가지 이상의 서비스 개발을 하게 되는 경우 시스템에서 개개인의 권한을 일일이 관리하는 것은 매우 비효율적이다. 또한, 권한의 변화가 자주 일어나는 규모가 큰 시스템에서 각각의 주체에 대하여 모든 권한의 기술이 힘들고 결국에는 사용자 접근제어 관리의 효율성 감소의 문제로 이어지게 된다.

역할 기반 접근제어(role-based access control, RBAC)[9]는 앞서 제시한 접근제어 모델들의 문제점을 보완하기 위한 방법으로 객체에 대한 권한을 역할이라는 추가적인 표현에 포함해 사용자가 역할이라는 멤버에 포함됨으로써 접근통제가 이루어진다. 하지만 강제적 접근제어와 같이 잦은 권한 변경으로 인하여 발생하는 문제로 사용자의 동적인 권한 변화에 대해서 효율적인 모델링이 불가능하다. 예를 들어 사용자가 새로운 매쉬업 서비스를 생성하기 위해 자신이 필요한 서비스가 담긴 역할을 선택하기 원한다. 만약 시스템에 정의되어 있지 않은 역할이라면 새로운 역할의 생성이 필요로 하게 된다. 이는 시스템 규모에 비례하여 계속 증가하게 되며 단순히 역할의 확장만으로는 사용자의 모든 요구에 대한 접근제어는 비효율적으로 변하게 되고 역할이 가지는 장점이 사라지게 된다[11].

본 논문에서는 기존의 접근제어 방법에서 존재하는 문제점을 보완한다. 서비스 매쉬업 시스템에 적합한 접근제어 설계를 위하여 기존의 역할 기반 접근제어의 개념과 동적인 역할 생성을 사용한다. 또한 잦은 권한의 변경을 가지는 환경에서 더욱 효과적인 접근제어 관리 방법을 위해 동적 역할 기반 접근제어(dynamic role-based access control, DRBAC) 모델 제안한다. 제안된 동적 역할 기반 접근제어 모델은 사용자에 의해 서비스가 정의되고

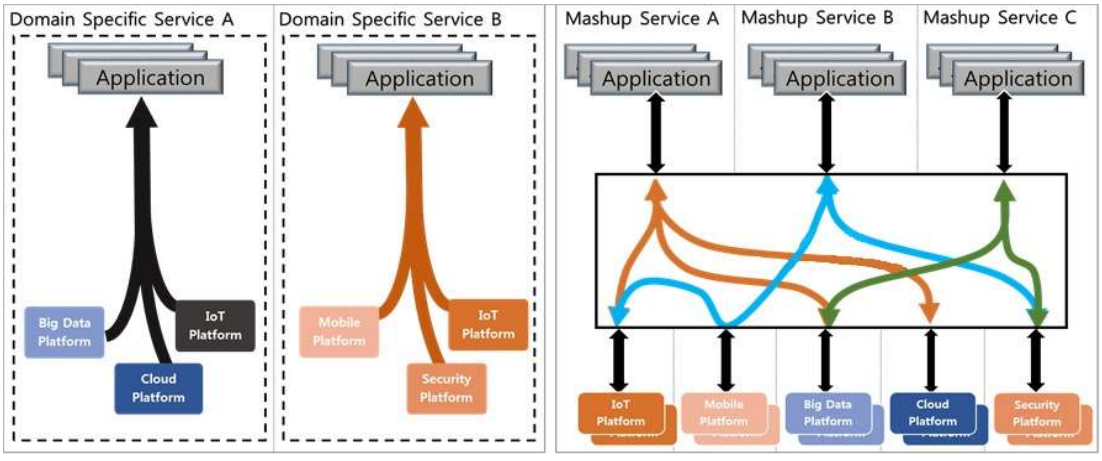

그림 1. 서비스 매쉬업 시스템
Fig. 1. Service Mashup System

그에 대한 권한을 생성과 함께 동적으로 할당받음으로써 기존의 접근제어 방법보다 역할-권한 할당이 동적인 상황에서 더욱 효과적인 접근제어를 수행할 수 있다.

Ⅱ. 관련 연구

1. 역할 기반 접근제어 모델

그림 2는 역할 기반 접근제어[9]의 표준 모델로서 강제적 접근제어와 임의적 접근제어의 문제점을 보완하기 위한 모델이다. 객체에 대한 권한을 역할이라는 추상적인 표현에 포함해 이를 사용자에게 분배하여 통제함으로써 동적인 환경 변화에서도 유동적인 대처가 가능한 장점을 가진다. 역할 기반 접근제어는 $RBAC_0$, $RBAC_1$, $RBAC_2$, $RBAC_3$의 4가지 모델로 이루어져 있으며 $RBAC_0$는 RBAC의 기본 모델로서 다양한 시스템에 적용할 수 있으며, $RBAC_1$는 역할로부터 권한을 상속받을 수 있는 역

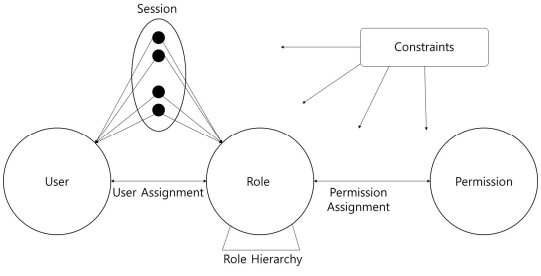

그림 2. RBAC 표준 모델[9]
Fig. 2. RBAC Reference Model[9]

할 계층에 대한 사항이 추가되었다. $RBAC_2$는 RBAC을 구성하는 각각의 요소에 제한 설정이 가능한 제약조건이 추가 되었다. $RBAC_3$는 $RBAC_0$, $RBAC_1$ 그리고 $RBAC_2$를 모두 통합한 모델로서 역할계층에 대한 제약조건의 설정이 가능하다.

2. 표준 참조 역할 기반 접근제어 모델

표준 참조 역할 기반 접근제어 모델[12]은 Sandhu의 역할 기반 접근제어 표준 모델[9]을 강화하여 기본형 역할 기반 접근제어 모델(Flat RBAC), 계층형 역할 기반 접근제어 모델(Hierarchical RBAC), 제한형 역할 기반 접근제어 모델(Constrained RBAC) 그리고 대칭형 역할 기반 접근제어 모델(Symmetric RBAC) 총 4가지로 구성하였다. 우선 제안된 모델 중 가장 기초가 되는 모델인 기본형 역할 기반 접근제어 모델은 사용자(user), 역할(role), 권한(permission)을 중심으로 이루어져 있으며 이러한 구성의 기본 개념은 사용자는 역할을 할당받을 수 있으며, 주로 역할은 권한들의 추상적인 표현으로 사용되며 권한을 할당받고 사용자에게 할당된 역할을 통하여 권한의 제어가 이루어진다. 사용자-역할의 관계와 권한-역할의 관계에서 각각의 관계들은 다대다의 관계를 맺게 된다. 이러한 관계를 통해 사용자는 동시에 여러 가지 역할을 가질 수 있다. 역할 또한 여러 가지 권한을 가질 수 있으며 사용자는 세션을 통하여 동시에 자신이 원하는 역할을 사용할 수 있다. 표 1은 NIST에서 정의한 역할 기반 접근제어 표준 참조모델의 설명과 계층화 및 제약조건에 대하여 간단하게 표로 나타낸 것이다.

표 1. NIST에서 제안한 역할 기반 접근제어 모델별 특징[12]
Table 1. Features of RBAC models proposed by NIST[12]

Model	RBAC Functional Capabilities	Hierarchy	Constraints
Flat RBAC	Contains basic RBAC functionality	X	X
Hierarchical RBAC	(Flat RBAC) + (Supports role hierarchy)	O	X
Constrained RBAC	(Hierarchical RBAC) + (Enforce separation of duty)	O	O
Symmetric RBAC	(Constrained RBAC) + (Support Permission-role review)	O	O

계층형 역할 기반 접근제어 모델은 앞서 말한 기본형 역할 기반 접근제어 모델을 바탕으로 역할 간의 계층구조 지원을 추가한 형태의 모델이다. 계층구조는 역할에 적용되며 계층구조에서 상위에 있는 역할은 하위에 있는 역할의 권한을 소유할 수 있으며 상속 또한 가능하다. 역할계층은 일반적 계층과 제한된 계층 2가지 형태의 서브계층을 갖는다.

제한형 역할 기반 접근제어 모델은 계층형 역할 기반 접근제어 모델을 바탕으로 임무분리의 지원을 추가한 형태로서 사용자의 역할 할당과 함께 사용자 간의 이해관계 충돌을 막기 위하여 사용자가 자신의 직위나 직책을 벗어나는 권한을 제한하여 불법적인 행위를 막을 수 있도록 도와준다. 이는 정적 임무분리와 동적 임무분리로 분류되며, 정적 임무분리는 동일 사용자가 정적 임무분리가 선언된 두 개 이상의 사용 역할에 배정되지 못하게 함으로 임무 임무분리가 적용되고, 동적 임무분리란 동일 사용자가 동적 임무분리가 선언된 두 개 이상의 역할에 배정될 수 있지만 동시에 이 역할들을 사용할 수는 없음을 나타낸다. 이런 식으로 적절한 제약을 추가함으로써 기존 계층형 역할 기반 접근제어 모델에서 나타나는 불법적인 행위에 대한 문제점을 해결한다.

대칭형 역할 기반 접근제어 모델은 제한형 역할 기반 접근제어 모델을 바탕으로 사용자 배정 제약조건과 유사한 권한 배정 제약조건을 추가한 형태이다. 권한-역할 뿐만 아니라 사용자-역할의 배정에서도 제약조건을 적용함으로 상황 변화에 따른 적절한 권한-역할 배정이 가능하다.

Ⅲ. 동적 역할 기반 접근제어 모델

이 장에서는 서비스 매쉬업 시스템에 효율적인 접근제어를 위해 동적으로 생성되는 역할을 바탕으로 동적 역할 기반 접근제어 모델을 제안한다. 본 논문에서 제안하는 모델을 그림 3과 같이 도식화하였다. 기존의 역할 기반 접근제어 모델에서 사용되는 구성요소와 함께 서비스와 권한 활성화 테이블이라는 새로운 구성요소를 추가하여 사용자에 의해 생성된 역할을 동적으로 등록할 수 있다.

동적 역할 기반 접근제어 모델의 구성은 표 2와 같다. 사용자는 시스템의 사용을 위한 사람을 의미하며, 역할은 접근제어 정책을 구현하는 직무 또는 직책을 의미한다. 권한은 하나 혹은 그 이상의 자원에 대한 접근권한을 나타낸다. 사용자는 역할의 멤버가 됨으로써 시스템 자원에 대한 권한을 획득할 수 있다. 역할 계층이란 역할의 상속관계를 의미하며 제약조건은 구성요소들에 조건 혹은 제한사항을 설정함으로써 사전에 발생할 수 있는 의무분리에 대한 문제점의 해결이 가능하다. 서비스란 동적인 역할 생성을 위해 시스템에서 제공되는 Open API들로 구성되며 사용자는 서비스를 통하여 새로운 역할의 생성이 가능하다.

새롭게 제안하는 동적 역할 기반 접근제어 모델은 서비스 매쉬업 시스템과 같이 다양한 종류의 서비스를 제공하는 시스템에서 사용자가 원하는 요구에 따른 동적인 권한 할당을 통하여 기존의 접근제어 모델들이 가지는 규모가 큰 시스템에서 발생하는 접근제어의 복잡성에 대한 문제를 해결한다. 동적 역할 기반 접근제어는 사용자가 동적으로 생성된 역할을 사용하기 위한 구성요소 중 하나인 서비

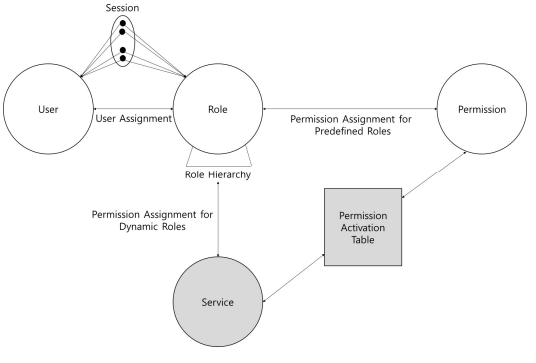

그림 3. 동적 역할 기반 접근제어 모델 구성도
Fig. 3. Diagram of the proposed DRBAC model

스와 권한 활성화 테이블을 거쳐 사용자에게 역할을 제공한다. 이는 역할 계층을 통해 사용자의 하위 역할로 동적 생성이 이루어지도록 한다. 또한 서비스 매쉬업 시스템의 정책에 따라 미리 구성된 역할들을 통해 시스템의 접근제어를 위한 사용자 역할 할당 및 권한 부여도 가능하다. 권한 활성화 테이블에 등록된 정책에 따라 역할별 권한이 정의되어 있으며 각각의 서비스들은 사용자의 요청에 따라 동적으로 등록된다.

Ⅳ. 서비스 매쉬업 시스템을 위한 동적 역할 기반 접근제어 모델 설계

이 장에서는 서비스 매쉬업 시스템의 접근제어를 위해 제안된 동적 역할 기반 접근제어 모델의 정의를 내린다. 동적 역할 기반 접근제어 모델의 새로운 개념과 예상 시나리오를 통하여 사용자에 의한 역할의 동적인 생성 과정을 표현한다. 서비스 매쉬업 시스템은 기존의 단일 서비스들을 조합하여 사용자의 다양한 요구에 맞는 새로운 서비스 개발을 위한 시스템이다. 표 2는 동적 역할 기반 접근제어 모델에 대한 개체의 정의를 나타내며, 제안된 모델에 대한 자세한 내용은 다음과 같다. 사용자는 서비스 매쉬업 시스템을 사용하기 위한 사용자의 이름으로 구성되어 있으며 역할은 크게 정적 역할(predefined role)과 동적 역할(dynamic role) 두 가지로 구분된다. 정적 역할은 관리자 혹은 시스템에 의해 이미 정의되어 있는 역할을 의미하며 서비스 매쉬업 시스템을 관리하고 전체 서비스의 권한을 가지는 시스템 관리자와 서비스 매쉬업 시스템에서 제공하는 서비스들을 개발하는 시스템 개발자가 이에 속한다.

표 2. 동적 역할 기반 접근제어 모델 구성
Table 2. Components of the DRBAC model

Component	Definition
User	System users
Role	Job function or title which defines an authority level
Permission	Access to resources
User Assignment	Assign users to roles
Permission Assignment	Assign permission to roles
Role Hierarchy	Role inheritance relationship
Constraints	Configuration Component or constraint
Service	Object registration list for dynamically created roles
Permission Activation Table	Specified permission and permission tables

동적 역할은 사용자의 요청에 따라 동적으로 사용되고 등록되는 역할로서 매쉬업 서비스 개발을 위한 서비스 개발자와 매쉬업 서비스에 사용될 Open API를 테스트해볼 수 있는 서비스 테스터가 이에 속한다. 사용자는 동적 역할을 부여받은 후 시스템을 통하여 자신이 사용할 서비스를 설정한다. 이후 설정된 서비스는 동적으로 사용자에게 역할 등록이 이루어지게 된다. 또한, 시스템 개발자가 새로운 서비스를 등록할 경우 Open API의 업데이트를 통해 Open API Pool에 등록되고 등록된 서비스는 동적 역할 생성이 이루어져 사용할 수 있게 된다.

이러한 과정을 나타내기 위해 서비스 매쉬업 시스템의 예상 시나리오 중 하나인 서비스 개발자의 캠퍼스 안전관리 서비스 개발이 있다. 캠퍼스 안전관리 서비스란 출입자 인식과 출입자 정보를 비교하여 관리자에게 통지함으로써 캠퍼스의 안전을 관리하는 출입자 관리 서비스이다. 우선 매쉬업 서비스 개발을 위해 사용자는 새로운 서비스를 등록을 한다. 그리고 등록된 서비스의 권한 활성화를 통해 자신이 가지는 역할(서비스 개발자, 서비스 테스터)을 정의하고 서비스와 역할별 권한이 기술되어 있는 권한 활성화 테이블과 대응된다. 마지막으로 대응된 결과에 따라 사용자에게 역할 권한 부여가 이루어지게 된다. 여기서 권한 활성화 테이블이란 역할과 역할의 권한이 정의되어있는 테이블 구조로 이루어져 있다. 권한에서 의미하는 서비스와 동작의 개념은 시스템에서 제공되는 모든 서비스와 서비스 사용을 위한 API 호출, 서비스의 조합을 위한 API 매쉬업, 서비스의 테스트를 위한 API 테스트 등과 같이 서비스의 동작을 의미한다.

역할의 권한 부여는 서비스와 권한 활성화 테이블을 통하여 서비스에서 동적으로 등록된 하위역할에 권한이 부여한다. 그림 4는 동적 역할에 기반을 둔 사용자의 접근제어를 위한 과정을 도식화한 것이다. 첫 번째는 서비스 매쉬업 시스템의 서비스 개발자인 사용자 A는 매쉬업 서비스 개발을 위하여

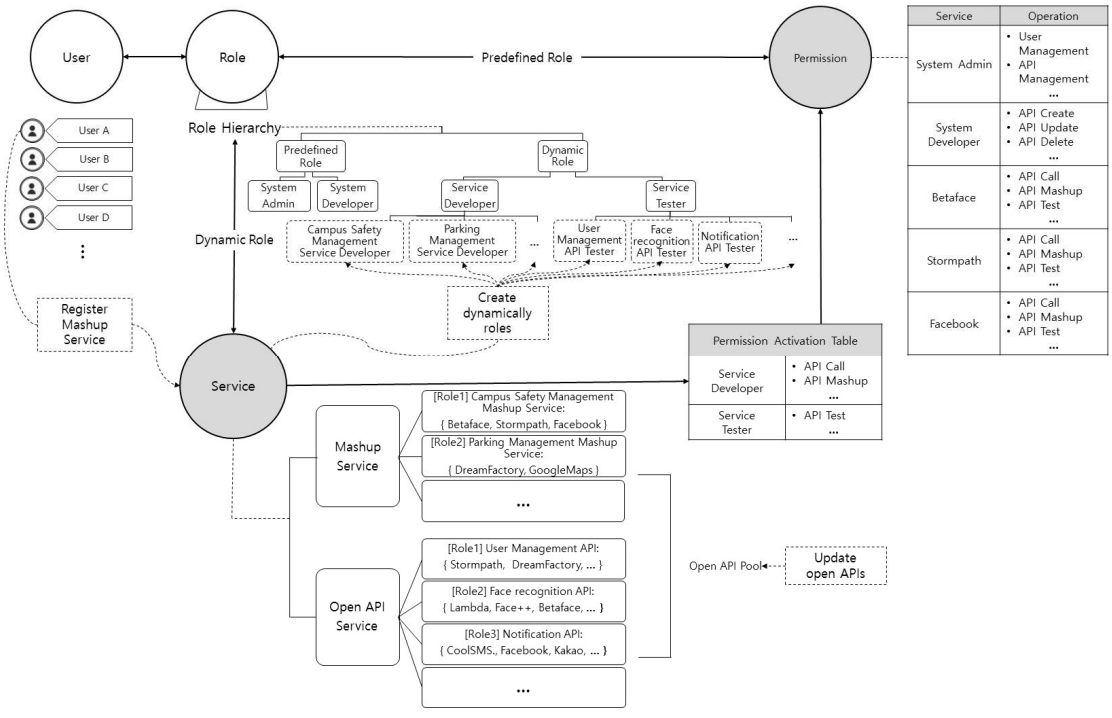

그림 4. 동적 생성 역할에 기반을 둔 사용자 접근제어
Fig. 4. User access control based on dynamic role generation

캠퍼스 안전관리 매쉬업 서비스를 등록하고 등록된 역할에 대한 권한을 획득하는 시나리오이다.

(1) 사용자 A는 새로운 매쉬업 서비스 개발을 위하여 서비스 매쉬업 시스템의 서비스 개발자 역할을 획득한다.

(2) 서비스 개발자 역할을 가진 사용자 A는 서비스의 매쉬업 서비스에서 자신의 개발에 필요한 얼굴인식 API인 Betaface와 사용자 관리 API인 Stormpath 그리고 통지 API인 Facebook Messenger를 선택한다.

(3) 선택된 서비스들을 캠퍼스 안전관리 서비스라는 이름으로 등록하게 되면 자신의 하위 계층에 역할 동적 생성이 이루어진다.

(4) 사용자 A는 역할 정보와 선택된 서비스의 정보를 통하여 권한 활성화 테이블과 대응 된다. 이때 권한 활성화 테이블은 기존에 시스템에서 정의된 정책을 통하여 서비스 개발자는 API의 호출과 API의 매쉬업이 가능한 권한을 획득하게 된다.

(5) 권한 활성화 테이블에서 정의된 권한을 통해 등록된 캠퍼스 안전관리 서비스 역할에 권한이 부여되고 사용자 A는 자신이 가진 캠퍼스 안전관리 서비스 역할의 멤버가 되므로 캠퍼스 안전관리 서비스의 역할이 가지는 Betaface, Stormpath, Facebook Messenger의 API 호출, API 매쉬업의 권한을 획득한다.

다음 시나리오는 서비스 매쉬업 시스템의 서비스 테스터인 사용자 B가 매쉬업 서비스 개발에 사용할 얼굴인식 Open API를 시험해보고 결정하기 위해 얼굴인식 API 테스터 역할을 가지는 시나리오이다.

(1) 사용자 B는 자신이 매쉬업 서비스에 개발할 얼굴인식 API의 테스트를 위해 Open API 사용 요청을 한다.

(2) 요청된 Open API인 얼굴인식 기능이 포함된 Open API Pool의 얼굴인식 API는 요청에 따라 사용자 B의 하위 계층 역할로 등록된다.

(3) 등록된 역할과 권한 활성화 테이블에 정의된 권한을 획득함으로 사용자 B는 시스템에서 제공되는 모든 얼굴인식 API를 테스트할 수 있는 권한을 획득한다.

본 논문에서 제안하는 동적 역할 기반 접근제어 모델을 적용한 두 가지 시나리오와 같이 개발자들은 서비스 매쉬업 시스템에서 기존의 Open API 서비스들을 활용하여 새로운 매쉬업 서비스를 개발하게 된다. 이러한 상황에서 개발자들은 자신이 개발

을 원하는 서비스를 선택하고 그에 따라 동적 역할이 자동으로 생성되고 해당 개발자에게 할당됨으로써 개발을 위한 API 테스트와 API의 사용 권한을 얻을 수 있다. 이를 통하여 기존의 정적인 역할 기반 접근제어를 사용하였을 때 발생하는 역할 확장에 대한 비효율성과 서비스 매쉬업 시스템에 적용하였을 때 발생하는 역할 관리의 복잡성 문제를 해결할 수 있다.

V. 결론 및 향후 연구

서비스 매쉬업 시스템은 사용자의 역할에 따라 동적인 권한의 요구가 발생하게 된다. 이러한 요구는 기존에 존재하는 접근제어 모델들이 서비스 매쉬업 시스템과 같이 다양한 사용자의 요구에 따른 다양한 변화를 가지는 시스템에서 사용하게 될 경우 시스템의 접근제어에 대한 복잡성 증가와 효율성의 감소 문제점을 발생시켰다. 본 논문에서는 앞서 발생한 서비스 매쉬업 시스템에서 존재하는 접근제어의 문제점을 해결하고자 사용자의 역할과 요구에 따라 동적인 역할 변경을 기존의 역할 기반 접근제어 개념과 서비스, 권한 활성화 테이블을 통해 접근제어가 가능한 동적 역할 기반 접근제어 모델을 제시하였다. 향후 연구로는 제안한 동적 역할 기반 접근제어 모델을 현재 개발 중인 서비스 매쉬업 시스템에 직접 적용하여 접근제어 확장성에 대한 실험을 진행하는 것이다.

References

[1] J. Yu, B. Benatallah, and F. Casati, "Understanding Mashup Development," *IEEE Internet Computing*, vol. 12, issue 5, Oct. 2008.

[2] D. Lee, S. Jeong, T. Jeong, and W. Hong, "Study of Smart Mediator for ICBMS platform interworking and mashup service development," *in Proc. KNOM Conference 2016*, pp. 71-75, Gangwon, Korea, May 2016.

[3] D. Lee, S. Jeong, T. Jeong, J. Yoo and W. Hong, "ICBMS SM: A Smart Mediator for Mashup Service Development," *in Proc. Asia-Pacific Network Operations and Management Symposium 2016*, pp. 1-6, Kanazawa, Japan, Oct. 2016.

[4] I. Lim, H. Kim, and H. Ju, "A RBAC-based campus access control method," *in Proc. KNOM Conference 2016*, pp. 109-110, Gangwon, Korea, May 2016.

[5] H. Mun and K. Han, "A Study on Design for Efficient Personal Policy of Service based RBAC," *Journal of Digital Convergence*, vol. 14, no. 2, pp. 191-196, Feb. 2016.

[6] H. Mun and J. Suh, "Sensitive Personal Information Protection Model for RBAC System," *Journal of The Korean Society Of Computer And Information*, vol. 13, no. 5, pp. 103-110, Sept. 2008.

[7] R.S. Sandhu and Q. Munawer, "The ARBAC99 model for administration of roles," *in Proc. Computer Security Applications Conference 1999.* pp. 229-238, Scottsdale, AZ, USA, Dec. 1999.

[8] R.S. Sandhu and P. Samarati, "Access Control: Principles and Practice," *IEEE Communications Magazine*, vol. 32, issue 9, Sept. 1994.

[9] R.S. Sandhu, E.J. Coyne, and H.L. Feinstein, "Role-Based Access Control Models," *IEEE Computer*, vol. 29, issue 2, pp. 38-47, Feb. 1996.

[10] W. Kim and C. Lee, "Design of RBAC Applying Efficient Separation of Duty," *Korea Computer Congress*, vol. 34, no. 1, pp. 44-49, June 2007.

[11] I. Lim, D. Kwon, H. Kim, D. An and H. Ju, "Design of User Access Control Model based on RBAC for Service Mash-up System", *in Proc. KIISE Winter Conference,* pp. 879-881, Gangwon, Korea, Dec. 2016.

[12] R. Sandhu, D. Ferraiolo, and Kuhn. R, "The NIST Model for Role-based Access Control: Towards a Unified Standard," *in Proc. ACM Workshop on Role Based Access Control 2000,* pp. 47-64, New York, NY, USA, July 2000.

임 이 삭 (Isaac Lim)

2015년 8월: 계명대학교 컴퓨터공학과 학사

2015년 9월~현재: 계명대학교, 컴퓨터공학과 석사과정

〈관심분야〉 사용자 접근제어 및 권한 관리, 네트워크 및 보안

권 동 우 (Dongwoo Kwon)

2010년 2월: 계명대학교 컴퓨터공학과 학사
2012년 2월: 계명대학교 컴퓨터공학과 석사
2013년 9월~현재: 계명대학교 컴퓨터공학과 박사과정
<관심분야> 미디어 스트리밍, 인터넷 침입예측, 네트워크 관리 및 보안

안 동 혁 (Donghyeok An)

2006년 2월: 한동대학교 전산전자공학부 학사
2013년 2월: 한국과학기술원 전산학과 박사
2014년 8월: 성균관대학교 컴퓨터공학과 초빙교수
2015년 2월: 삼성전자 책임연구원
2015년 3월~현재: 계명대학교 컴퓨터공학부 조교수
<관심분야> 유무선 네트워크, IoT, 콘텐츠 중심 네트워크

주 홍 택 (Hongtaek Ju)

1989년 8월: 한국과학기술원 전자계산학과 학사
1991년 8월: 포항공과대학교 컴퓨터공학과 석사
1997년 8월: 대우통신종합연구소 선임연구원
2002년 2월: 포항공과대학교 컴퓨터공학과 박사
2002년 9월~현재: 계명대학교 컴퓨터공학과 교수
<관심분야> 네트워크 및 시스템 관리, IoT 관리, SDN 네트워크 관리, 인터넷 침입 예측, 네트워크 보안