웹 기반 응용을 위한 직무 기반 접근 제어 모델의 설계

이 호*
국립한국재활복지대학 정보보안과 부교수

요 약

접근제어는 컴퓨터 운영체제, 워크플로우 시스템, 정보 보안 시스템 등에서 보편적으로 사용하고 있는 기술이다. 본 논문에서는 보안성 무결성 및 흐름제어 보안 기능을 제공하면서 동시에 직무 중심 조직의 접근 제어 요구를 용이하게 수용할 수 있는 직무 기반 접근 제어 모델을 제안한다. 제안한 접근제어 모델은 주로 웹을 기반으로 하는 응용 시스템에서 간단하면서도 안전한 방법으로 직무 기반 접근 제어를 수행하도록 설계되어있다.

Design of a Role-Based Access Control Model for Web-based Applications

Lee, Ho*

ABSTRACT

The access controls are the methods which are generally used in such systems as computer operating systems, workflow systems, information security systems and etc.. In the paper, is proposed a role-based access control model which not only has fundamental security functions such as security, integrity and flow control, but also meets the access control requirements of role-based social organizations. The proposed role-based access control model is designed in order to perform its functions in simple and secure way, largely in the environment of web-based applications.

Ⅰ. 서 론

직무 기반 접근 제어는 신분 기반 및 규칙 기반 접근 제어[1]의 특성을 포함하고 있는 방법으로 볼 수 있는데, 개인의 신분이 아닌 자신이 수행하는 직무에 따라 접근할 수 있는 정보의 종류와 사용할 수 있는 정보의 범주가 정해진다.

규칙 기반 접근 제어가 신분 기반 접근 제어의 완전한 대체 방법이 아니듯이 직무 기반 접근 제어[2]도 신분 기반 접근 제어와 규칙 기반 접근 제어의 병합이 아닌 상호 보완적인 방법이다. 따라서 이러한 세 가지 기존의 접근 제어 방식에 근간을 둔 새로운 접근 제어 모델의 설계가 가능할 수 있다.

본 논문에서는 검증된 기존의 접근 제어 메커니즘을 수용하여 보안성, 무결성 및 흐름제어 보안 기능을 제공하면서 동시에 직무 중심 조직의 접근 제어 요구를 용이하게 수용할 수 있는, 실제적인 접근 제어 시스템에 적용이 가능한 직무 기반 접근 제어 모델을 제안한다.

Ⅱ. 모델 설계

2.1 접근 제어 정보의 구성

주체에 있어서, 직무는 실체가 수행할 수 있는 역할을 말한다. 직무는 응용 분야에 따라 고유한 직무를 정의할 수 있으며 각각의 직무는 서로 다른 권한을 부여 받을 수 있다. 각 직무 식별자와 이에 따른 보안성 등급, 무결성 등급이 주체의 접근 제어 정보에 명시된다(표1 참조).

객체에 있어서, 식별자는 시스템에서 주체가 객체에 접근을 시도할 때 객체를 식별할 수 있도록 하는 식별자이다. 소유권자는 객체의 소유권을 가지는 실체를 나타내는 식별자로서 여기서는 특정한 직무이다. 각 객체 식별자와 이에 따른 보안성 등급, 무결성 등급 및 소유권자가 객체의 접근 제어 정보에 명시된다[6](표2 참조).

<표 1> 주체 접근 제어 정보
Table. 1 Subjects' access control info.

직무 식별자	보안성 등급	무결성 등급
r1	Top Secret	Crucial
r2	Secret	Very Important
r3	Secret	Important
r4	Confidential	Important

<표 2> 객체 접근 제어 정보
Table. 2 Objects' access control info.

객체 식별자	보안성 등급	무결성 등급	소유권자
o1	Top Secret	Crucial	r1
o2	Secret	Important	r2
o3	Secret	Very Important	r3
o4	Confidential	Important	r2

2.2 접근 제어 규칙 표기법 정의

집합 표현에서의 대소문자는 집합과 원소를 나타내며, 함수 표현에서의 매개 변수는 실체를 나타낸다.

S : 접근 제어의 주체 집합, $s \in S$

O : 접근 제어의 객체 집합 , $o \in O$

P : 접근 허가 집합, P=c, r, w, x, d $p \in P$,
(c :create, r :read, w :write,
x :execute, d :delete)

S_Level(a) : 보안성 등급 함수

I_Level(a) : 무결성 등급 함수

permit(role, a) : role의 a에 대한 접근 권한
검색 함수
owner(a) : 소유권자 함수
get_ACI(a) : ACI 요구 함수
exist_ACI(a) : ACI에 대해서 a의 존재 여부 확인 함수

2.3 접근 제어 규칙 정의[5,7]

규칙 1 : 주체(s)가 객체(o)에 대하여 수행하고자 하는 접근 허가(p)가 있을 때 주체와 객체는 ACI에 존재해야 하며, 주체가 객체에 대해 요구한 접근 허가가 부여될 수 있는 접근 제어 정보가 명기되어 있어야 한다.

```
role_acr(s, o, p)
{
  if ( exist_ACI(s)  and  exist_ACI(o) )
  then
    role ← get_ACI(s)
    if  ( p == permit(role, o) )  then
      return TRUE
    else if
      return FALSE
    endif
  else if
    return FALSE
  endif
}
```

규칙 2 : 보안성 보장을 위하여 주체의 보안성 등급이 객체의 보안성 등급을 지배할 때는 read와 execute 권한을 허용하고, 주체의 보안성 등급이 객체의 보안성 등급과 동일할 때는 *create*와 *delete*권한을 허용한다. 무결성 보장을 위해서는 주체의 무결성 등급과 객체의 무결성 등급이 일치할 때만 create, write, execute 및 delete 권한을 허용한다.

```
rule_acr(s, o, p)
{
  case p = 'c'
      if ( S_Level(s) == S_Level(o)  and
        I_Level(s) == I_Level(o) )
    then
      return TRUE
    endif
  case p = 'r'
      if ( S_Level(s) ≥ S_Level(o)  and
        I_Level(o) ≥ I_Level(s) )
      then
        return TRUE
      endif
  case p = 'w'
      if ( s == owner(o)  and
        S_Level(s) == S_Level(o)  and
        I_Level(s) == I_Level(o) )
      then
          return TRUE
      endif
  case p = 'x'
    if ( S_Level(s) ≥ S_Level(o)  and
        I_Level(s) == I_Level(o) )
    then
        return TRUE
      endif
  case p = 'd'
      if ( s == owner(o)  and
       S_Level(s) == S_Level(o)  and
       I_Level(s) == I_Level(o) )
      then
        return TRUE
      endif
```

```
    otherwise
      return FALSE
}
```

규칙 3 : 상위의 보안성 등급을 가진 객체가 하위의 보안성 등급을 가진 객체로 상위 보안성 등급의 정보를 불법적으로 전달해 주는 것을 방지하기 위해서 흐름제어가 필요한데, 주체(s)가 한 객체가 소유한 정보(o1)를 다른 객체(o2)로 전달하기 위한 동작을 하기 위해서는 주체의 두 객체에 대한 보안성 등급이 지배 관계에 있어야 하고 두 객체간에는 보안성 등급과 무결성 등급이 일치해야만 한다.

```
flow_acr(s, o1, o2)
{
   if ( s == owner(o1)  and
      S_Level(s) ≥ S_Level(o1)  and
      S_Level(s) ≥ S_Level(o2)  and
      S_Level(o1) == S_Level(o2)  and
      I_Level(o1) == I_Level(o2) )
   then
       return TRUE
   else if
      return FALSE
   endif
}
```

2.4 접근 제어 규칙 적용 모델

앞 절에서 설계한 접근 제어 규칙들을 적용하여 동작하는 규칙 적용 모델을 그림1에서와 같이 구성한다[6]. 이 모델은 주체의 객체에 대한 접근 요구가 발생하는 시점에 실시간으로 적용하도록 되어 있다.

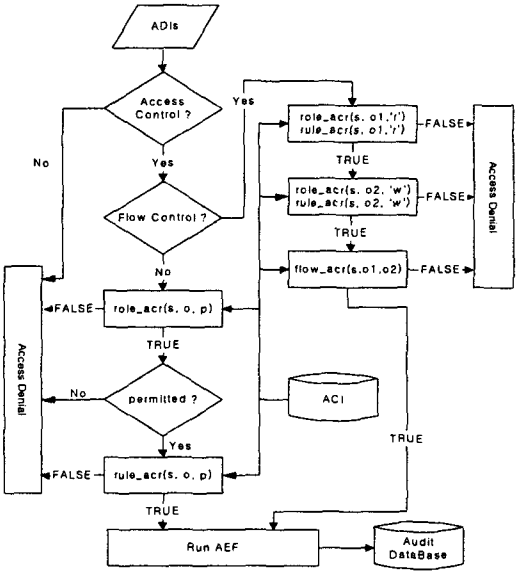

그림 1 접근 제어 규칙 적용 모델

Fig. 1 Model of applying access control rules

III. 기존 접근제어 방식과의 메커니즘 비교

본 논문에서 제안한 직무 기반 접근 제어 모델(이하 PRM이라 칭함)은 UNIX 처럼 전통적으로 사용자를 접근 제어의 주체로 하는 운영체제나 최근의 Windows 2000처럼 사용자와 그룹을 접근 제어의 주체로 하는 운영 체제들이 사용하는 접근 제어 방식과 비교하여 다음의 차이점을 갖는다.

첫째, UNIX와 Windows 2000은 파일 시스템이 제공하는 이미 구성되어 있는(precompiled) ACL 등의 접근 허가 정보를 이용하여 정적인 접근 제어를 수행하지만 PRM은 필요한 시점(run-time)에 접근 제어 규칙을 적용함으로써

접근 허가를 결정짓는 동적인 접근 제어 방법을 사용한다.

둘째, UNIX와 Windows 2000의 접근 제어 리스트(ACL)나 보안 서술자(SD)와 같은 접근 제어 정보는 많은 기억 공간을 필요로 하지만 [3] PRM은 접근 제어 규칙 적용의 근거가 되는 주체와 객체의 최소한의 보안 정보만(주체와 객체의 ACI만) 유지하면 되므로 많은 기억 공간을 필요로 하지 않는다.

셋째, Windows 2000 조차도, 최근의 접근 제어 기술로서 그 관리의 편리성 때문에 인정을 받고 있는 직무 기반 접근 제어 측면에서 보면, 직무 기반 접근 제어의 특성 중에 하나인 상위 계층으로의 상속을 반영치 못하고 있으나, PRM은 통상적인 직무 기반 접근 제어의 메커니즘으로 동작하므로 상위 계층의 보안 정보가 하위 계층의 보안 정보를 상속할 수 있다.

넷째, UNIX나 Windows 2000 운영체제의 접근 제어 모드는 파일 시스템이 제공하는 모드에 근거하여 결정되지만 PRM의 경우는 모델이 갖고있는 접근 제어 규칙에 따라 접근 제어 모드가 정해진다.

다섯째, UNIX나 Windows 2000 운영체제의 접근 제어는 범용성의 일반적인 컴퓨팅 환경에서 객체에 대한 접근 제어 요구를 수용할 수 있도록 되어있는데 비하여, PRM은 웹 기반 응용 시스템이라는 웹 서버 상에서 동작하는 시스템의 접근 제어를 목적으로 설계된 특수 목적의 접근 제어 모델이다.

여섯째, PRM은 보안이 일반적으로 취약한 웹 기반 응용 시스템을 위하여 제안된 엄격한 보안 규칙을 적용하는 안전한 접근 모델이므로, 파일 시스템의 기능을 이용하여 객체별로 복잡한 접근 제어를 수행하는 UNIX나 Windows 2000의 방법과 비교하면, 간단하지만 확실한 보안 체계를 제공한다.

Ⅳ. 기존 접근 제어 방식과의 성능 비교

여기서는 접근 제어 정보(ACI) 만을 Windows 2000의 경우와 비교하고자 한다.

4.1 주체의 접근 제어 정보[7,8]

주체의 접근 제어 정보를 표시하기 위하여 PRM은 주체의 ACI를 사용하고 Windows 2000은 액세스 토큰을 사용한다[3]. 표3과 표4는 각각 하나의 주체를 위한 ACI와 액세스 토큰의 구조를 보여준다. PRM의 경우에는 사용자가 복수의 직무를 부여받는 경우도 있지만 주체 ACI를 기준으로 한다면 모든 직무는 ACI에 하나의 엔트리로서 표시된다. Windows 2000의 경우에는 사용자 별로 복수의 그룹 보안 식별자 및 특권이 부여될 수 있으므로 주체의 접근 제어 정보를 표시하기 위하여 PRM의 경우보다는 많은 양의 기억 공간과 복잡한 데이터 구조가 필요하다는 것을 알 수 있다. Windows 2000의 경우에 사용자가 여러 그룹에 속하는 경우에는 그에 따라 특권도 늘어나기 때문에 필요한 정보를 저장하기 위한 기억 공간이 그 만큼 더 요구되며 따라서 PRM의 경우와는 다르게 가변적인 크기의 데이터 구조를 사용해야 한다는 사실을 알 수 있다.

<표 3> 주체의 ACI
Table. 3 Subject's ACI

직무 식별자	보안성 등급	무결성 등급
r3	Top Secret	Crucial

<표 4> 주체의 액세스 토큰
Table. 4 Subject's access token

User SID	Group SIDs	Privileges
Jane User	Administrators Power Users Service Operators Users	Start Service Load Device Driver Shut Down

4.2 객체의 접근 제어 정보[7,8]

객체의 접근 제어 정보를 표시하기 위하여 PRM의 경우는 표5에서처럼 객체의 ACI를 사용하는데 Windows 2000은 그림2에서처럼 보안 서술자를 사용한다[3].

<표 5> 객체의 ACI
Table. 5 Object's ACI

객체 식별자	보안성 등급	무결성 등급	소유권자
o4	Top Secret	Crucial	r3

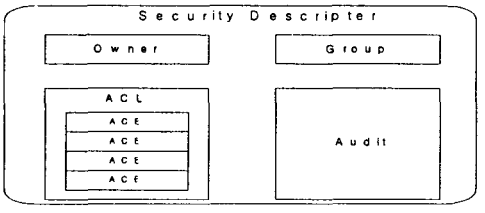

그림 2 객체의 보안 서술자
Fig. 2 Object's security descriptor

PRM의 경우에는 표5에서 보여 주듯이 모든 객체가 동일한 구조의 접근 제어 정보를 갖는다. 반면에 Windows 2000의 경우에는 그림2에서 보듯이 PRM의 경우보다는 아주 복잡한 데이터 구조를 가진다. 그림에서 보안 서술자를 구성하는 구성 요소 중에 ACL이 있는데 이 ACL은 다시 임의 개수의 ACE로 구성된다. 그림2는 네 개의 ACE로 이루어진 ACL의 예를 보여준다. ACE는 객체를 접근할 수 있는 접근 허가가 주어진 주체, 접근 허가 여부 및 접근 허가 종류 등을 나타내는 정보를 포함하고 있다. 이 경우에 ACE는 사용자 또는 그룹과 같이 접근 허가를 부여 받는 주체별로 한 엔트리씩 존재한다. 즉 접근 허가를 부여 받은 사용자 및 그룹의 총 숫자 만큼의 ACE가 ACL에 존재한다는 것을 의미한다. 이 경우에도 주체의 접근 제어 정보에서와 같이, 접근 제어 정보를 구성하는 요소 정보를 표시하는 정보의 표시 단위 및 데이터 구조 등이 구현 방법에 따라 다를 수 있기 때문에 정확한 정량적인 비교는 어렵고, 객체의 접근 제어 정보의 양이 PRM 의 경우는 항상 일정한 크기이므로 이것을 1로 보았을 때 Windows 2000 의 경우에는 대략 객체 접근이 허가된 주체 전체(r, 여기서 r은 ACE의 총 개수와 동일함)의 k(PRM의 경우를 1로 보았을 때 $k > 1$) 배수(multiple) 이상의 접근 제어 정보 저장을 위한 기억 공간(S)이 필요하다. 이를 수식으로 나타내면 $S = kr$ ($r = m + n$, m은 사용자 수, n은 그룹 수) 이다. 이 경우에 보안 서술자의 소유자, 그룹 및 감사 정보 저장 공간은 계산에 산입하지 않았다. 이러한 비교치가 객체의 접근 제어에 필요한 정보량에 있어서 PRM이 Windows 2000 보다 ACI를 위한 기억 공간 효율성이 뛰어나다는 것을 증명한다.

V. 기존 접근제어 방식과 웹 기반 응용에서의 적합성 비교

클라이언트 서버 환경에서와 같이 일반적인 접근 허가를 필요한 경우에는 어느 사용자(주체)가 어떤 접근 권한을 사용하여 어느 객체에 접근할 수 있는지를 결정하도록 되어 있다.

반면에, 웹 기반 환경에서의 접근 허가는 웹 사이트를 방문하는 모든 사용자(주체)를 대상으로 하여 웹 서버의 특정 페이지를 볼 수 있는지, 특정 스크립트를 실행할 수 있는지, 웹 서버에 정보를 업로딩할 수 있는지 등의 객체에 대한 접근 권한을 통제하는 것이 필요하다[4,8]. 이 경우에 웹 서버에서는 웹 서버와 웹 클라이언트 사이에 생성된 세션(session)을 단위로 하여, 주체와 객체의 접근 제어 정보를 참조하여

접근 권한을 결정하도록 되어 있다.

작고 정형화된 표3과 표5에서와 같은 데이터 구조를 사용하여 그림1의 모델(정형화된 작은 프로그램으로 구현이 가능)을 실행하여 접근 권한을 판단하는 경우와 크고 동시에 가변 크기를 갖는 표4와 그림2에서와 같은 구조의 데이터를 사용하여 접근 권한을 판단하는 경우에는 어느 편이 보조 기억 장치에 있는 데이터의 접근 횟수가 많고 접근 장소 결정 방법이 복잡하냐에 따라 처리 성능이 좌우된다고 판단된다. 이 경우에 직무 기반 접근 제어에서는 주체의 접근 제어 정보가 직무 별로 만 존재하면 되므로 이를 주기억 장치 공간에 상주 배치할 수 도 있다.

본 논문에서 제안한 직무 기반 접근 제어 방식을 웹 기반 응용을 위해 사용한다면, 세션 별로 빈번히 발생하는 접근 권한 결정 문제를 ACI와 접근 제어 규칙을 이용하여 쉽고 안전한 방법으로 해결하는 것이 가능하다고 판단된다.

VI. 결 론

본 논문에서는, 보안 기능을 만족 시키면서도 효율적인 관리가 가능한 직무 기반 접근 제어 모델을 제시하기 위하여, 보안성 무결성 및 흐름제어가 가능한 접근 제어 규칙을 정의하였고 이를 토대로 규칙 적용을 위한 모델을 설계하였다.

기존의 접근 제어 방식과의 비교를 통하여, 제안한 직무 기반 접근 제어 방법이 주체와 객체의 보안 속성 정보를 근거로 보안 규칙을 적용하는 간단하면서도 안전한 방법임을 입증하였고 윈도우즈 2000의 접근 제어 방법과 비교하여 접근 제어 정보 저장을 위한 기억 공간의 효율성이 뛰어남 증명하였다.

기존 접근제어 방식과의 웹 기반 응용에 있어서의 적합성 비교에 있어서는, 세션 단위로 빈번한 접근 권한 판정이필요한 웹 환경에서 제안된 직무 기반 접근 제어 방법이 어떻게 효율적으로 활용될 수 있는가를 증명하였다. 단 여기서는 컴퓨터 분야에서 일반적으로 알려진 사실을 기초로 하여 효율성을 입증하였으며 계량화된 효율성 분석 문제는 별도의 연구 주제로 다룰 수 있을 것으로 판단한다.

참고문헌

[1] Warwick Ford, "Computer Communications Security - Principles, Standard Protocols and Techniques", Prentice Hall, pp. 149-176, 1994.

[2] Ravi S. Sandhu, Edward J. Coyne, "Role-Based Access Control Models", IEEE Computer, pp. 8-47, Feb., 1996.

[3] Michael M. Swift, "Improving the Granularity of Access Control in Windows NT ", In Proc. of ACM RBAC 2001, pp. 87-96, 2001.

[4] Elisa Bertino, Silvana Castano, Elena Ferrari, "On Specifying Policies for Web Documents with an XML-based Language ", *In Proc. of ACM RBAC 2001*, pp. 57-65, 2001.

[5] 이호, 정진욱, "안전한 인터넷 사용을 위한 접근 제어 메커니즘 설계", 한국 OA 학회 논문지, Vol. 5 No. , pp. 84-90, 2000.

[6] 이호, 정진욱, "직무-기반 접근 제어 모델 설계", 한국 OA 학회 논문지, Vol. 6 No. 1, pp. 60-66, 2001.

[7] 이호, "안전한 직무 기반 접근제어에 대한 연구", 한국 OA 학회 논문지, Vol. 6 No. 4, pp. 119-124, 2001.

[8] 이호 "웹 기반 응용 시스템을 위한 안전한 직무 기반 접근 제어 모델에 관한 연구", 박사 학위 논문, 성균관 대학교 대학원 정보공학과, 2002

이 호

1989년 벨기에 VUB 대학원 정보공학과 공학석사

2002년 성균관 대학교 대학원 정보공학과 공학박사

1982-1991년 한국전자통신연구원 선임 연구원

1991-2000년 대덕대학 정보통신과 조교수

2000-2002년 송호대학 정보산업계열 부교수

2002년 현재 국립 한국재활복지대학 정보보안과 부교수