2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

PBEAC : 무결성을 지원하는 목적지향 접근제어

차병래*, 서재현*
*목포대학교 컴퓨터공학과
e-mail:chabr@webs.mokpo.ac.kr

Purpose-oriented BEAC Supporting for Integrity

Byung-Rae Cha*, Jae-Hyun Seo*
*Dept of Computer Engineering, Mokpo National University

요약

목적지향 접근제어는 자율적 접근제어에 목적이라는 개념을 부가하여 강제적 접근제어의 기능을 추가할 수 있다. 본 논문에서는 주체와 객체 사이의 접근제어를 특정 목적에 부합한 경우에 접근이 가능하도록 하는 목적지향 접근제어 기능을 제공하도록 한다. 또한 분산 객체 기반의 접근제어를 제공하며 객체들 사이에 발생하는 복잡한 보안 요구사항들을 보안규칙으로 제공함으로서 객체들의 무결성이 유지되도록 하였다. 마지막으로 목적지향의 불리언 표연과 연산을 정의하고, 이를 이용한 PBEAC 모델을 제안하였다.

1. 서론

최근 인터넷의 급속한 발달에 따라 망을 안전하게 관리하기 위한 망 보안은 아주 중요한 분야로 대두되고 있다.

망 보안에는 정당한 사용자인지를 확인하는 인증 분야, 사용자가 객체를 사용할 수 있는지 여부를 확인하는 접근제어 그리고 사후감사를 위한 분야로 크게 구분할 수 있다. 접근 제어는 주체와 객체 상호작용에 어떠한 보안 속성을 할당하거나, 임의로 기술된 규칙에 의해 접근 허가를 평가하게 된다. 인터넷의 발달로 많은 사용자와 객체들 사이의 접근제어를 효율적으로 수행토록 하는 것은 아주 중요하다[3].

접근제어 정책은 자율적 접근제어와 강제적 접근제어 정책이 있는데 자율적 접근제어는 주체와 객체 사이에 어떤 유형의 연산이 수행가능한지 기술하는 것이고 강제적 접근제어는 주체에 인가등급을 부여하고 객체에 보안등급을 부여하여 이들 사이에 적절한 규칙을 제공하여 접근을 제어하는 정책이다. 최근 인터넷에서는 주로 자율적 접근제어를 사용하고 있는데 권한이 있는 사용자가 권한이 부여되지 않은 사용자 객체에 복사함으로서 발생하는 트로이 목마 같은 문제들이 발생한다. 그러므로 인터넷에서도 강제적 접근제어 정책이 수행되도록 해야한다[4].

또한 망 환경이 객체를 기반으로 하는 분산 컴퓨팅 환경으로 패러다임이 변화하고 있기 때문에 분산 객체를 기반으로 하는 접근제어 모델에 관한 연구가 필요하다. 분산객체를 지원하는 환경에서는 수학적 구조를 기반으로 하는 전통적인 접근제어 모델로는 복잡한 보안 요구사항들을 표현할 수 없다[2].

본 논문에서는 주체와 객체 사이의 접근제어를 특정 목적에 부합한 경우에 접근이 가능하도록 하는 목적지향 접근제어 기능을 제공하도록 한다. 또한 본 논문은 분산 객체 기반의 접근제어를 제공하며 객체들 사이에 발생하는 복잡한 보안 요구사항들을 보안규칙으로 제공함으로서 객체들의 무결성이 유지되도록 하였다. 마지막으로 BEAC 모델을 기반으로 목적지향 BEAC 모델을 제안하였다.

2. 관련연구

접근제어는 크게 자율적 접근제어와 강제적 접근제로 나뉘어 진다. 자율적 접근제어는 접근을 요청

한 사용자의 식별에 기반을 두고 있으며, 언제든지 사용자에게 특정 데이터에 대한 접근 권리를 부여하고 철회될 수 있기 때문에 자율적인 접근제어라 한다. 강제적 접근제어는 시스템의 데이터와 시스템 사용자가 여러 수준으로 비밀 분류되는 환경에서 강력한 보호가 요구되는 많은 정보가 존재하는 경우에 강제적 접근제어가 사용된다. 일반적으로 강제적 접근제어는 정보가 낮은 비밀 등급을 갖는 데이터로 흐르는 것을 방지하기 때문에 흐름 제어로도 정의된다[5].

BEAC(Boolean Expression based Access Control) 모델은 컴퓨팅 환경의 모든 개체를 주체와 객체로 양분한다. 주체의 범주를 정의하고 객체의 범주들을 정의하여 주체의 범주가 객체의 범주에 포함되는 경우에만 접근이 가능하도록 하였다. 이는 범주간의 관계를 표현하여 접근제어를 수행함으로서 강제적 접근제어의 기능을 수행한다[2].

목적지향 접근 규칙은 〈 s : u, o : t 〉의 형식으로 정의된다. 즉 주체 s가 객체 o를 연산모드 t로 접근할 때 u를 통해서만 접근하도록 하는 연구가 진행되었다[1].

본 논문에서는 주체와 객체 사이의 접근제어에 목적이라는 개념을 부가하여 특정 목적에 부합한 경우에만 접근이 가능하도록 강제적 접근제어 기능을 추가한다. 또한 분산객체 기반에서 목적지향 접근제어 기능을 제공하기 위한 불리언 표현과 연산을 정의하고, 이를 이용한 목적지향 BEAC 모델을 제안한다.

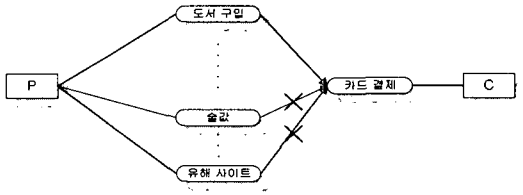
3. 목적지향 접근제어 모델

목적지향 접근규칙 $\langle o_i : op_i, o_{ij} : op_{ij} \rangle$의 목적 op_i는 o_i가 op_{ij}에 의해 o_{ij}를 조작하기 위한 연산으로 정의하고, o_i와 o_{ij}는 각각 부모와 자식 개체로 명명한다. 목적지향 접근규칙의 정의는 다음과 같다.

[목적지향 접근규칙 정의]

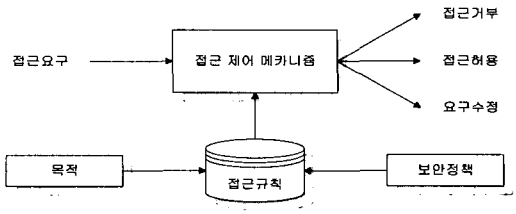
목적지향 접근 규칙 $\langle o_i : op_i, o_{ij} : op_{ij} \rangle$은 o_i의 op_i에 의해 파생된 연산 op_{ij}를 통해 o_i는 o_{ij}를 조작할 수 있다.

예를 들어, 개인 p는 p의 카드결제 객체 c로부터 카드결제를 시도한다고 가정하면, 목적지향 접근 규칙은 〈p:도서구입, C:카드결제〉형태이며, (그림 1)과 같이 목적지향 접근제어를 나타낸다.



(그림 1) 목적지향 접근제어의 개념

본 논문에서 제안되는 접근제어 시스템은 객체에 접근요구를 받아서 접근제어 모듈에서는 목적에 적합한 규칙이 존재하는 경우에는 접근이 허용되며 규칙이 존재하지 않으면 접근이 거부되며 경우에 따라서는 다른 목적을 적용하도록 요구수정이 발생하도록 한다.



(그림 2) 접근 제어 시스템의 구조

4. PBEAC: 목적지향 접근제어 규칙

BEAC 모델은 클라이언트/서버 컴퓨팅 환경의 모든 개체를 크게 주체와 객체로 양분하며, 각각의 그들 자체에 보안 속성을 갖고 있다. PBEAC (Purpose oriented BEAC)은 객체지향 시스템 컴퓨팅 환경에서는 모든 개체를 주체와 객체로 구분하지 않고 모든 것을 객체로 보는 시각을 제공한다.

PBEAC에서 연산 op_i의 보안속성 범주집합은 $CAT(op_i) = \{op_{i1}, op_{i2}, \cdots\}$이고, 객체 o의 보안속성 범주집합은 $CAT(o_i) = \{o_{i1}, o_{i2}, \cdots\}$이다. 또한, 불리언 표현을 사용하는 PBEAC 표현식 $P(op_i, o_j)$는 $CAT(op_i) = \{op_{i1}, \cdots\}$와 $CAT(o_i) = \{o_{i1}, \cdots\}$의 불리언 대수 연산("*": AND, "+":OR, "¯": 부정)에 의해 조합된 범주로 구성한다.

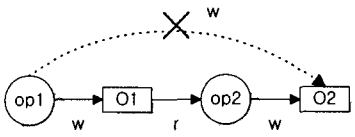
PBEAC 모델의 보안정책은 다음의 규약을 따른다.

- $o_1, o_2, \cdots$: 각각의 객체를 나타낸다.
- $CAT(op_i) = \{op_{i1}, op_{i2}, \cdots\}$: 연산 op_i의 범주 집합을 나타내며, 불리언 속성 값을 갖는다.
- $CAT(o_j) = \{o_{j1}, o_{j2}, \cdots\}$: 객체 o_j의 범주 집합을 나타내며, 불리언 속성 값을 갖는다.

- $(op_{i1}, m), (op_{i2}, m), \cdots$: 접근 모드 m을 갖는 연산 op_i의 범주 집합을 나타낸다. 일반적으로는 연산의 범주 집합에서 재사용 범주를 나타낸다.
- $P(op_i, o_j)$: $CAT(op_i)$과 $CAT(o_j)$의 불리언 대수 연산("*": AND, "+":OR, "¯": 부정)의 조합에 의한 목적을 나타낸다.
- $\widehat{op_i}$: 연산의 범주 집합에서 일회성 범주를 나타낸다.

목적지향 BEAC 모델의 무결성 보안제약조건은 이행 무결성, 상호 무결성 그리고 임무분리로 구분한다.

[Ⅰ] 이행 무결성 보안제약조건



(그림 3) 이행성 보안제약조건

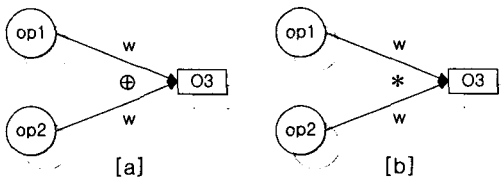
이행 무결성 보안제약조건은 연산 op_1의 범주 집합에는 op_{11}과 op_{12}로 구성되었으며, 각각의 연산의 접근 모드는 w, r, w 로 설정되어 있다. 또한 객체 o_1의 범주 집합에는 o_{11}과 o_{12}로 구성되었다. op_1과 o_1의 목적은 연산 op_{11}가 객체 o_{11}을 쓰기 연산을 수행하고, 이어서 연산 op_{12}가 객체 o_{11}을 읽기 연산을 수행한 다음, 연산 o_{12}가 객체 o_{12}을 쓰기 연산을 수행한다.

$CAT(op_1) = \{(op_{11}, w), (op_{12}, r), (op_{12}, w)\}$,

$CAT(o_1) = \{o_{11}, o_{12}\}$,

$P(op_1, o_1) =$
$\langle\{(op_{11}, w), o_{11}\}, \{(op_{12}, r), o_{11}\}, \{(op_{12}, w), o_{12}\}\rangle$

[Ⅱ] 상호 무결성 보안제약조건



(그림 4) 상호배제[a]와 임무분리[b] 보안제약조건

상호 무결성 보안제약조건은 연산 op_1의 범주 집합에는 op_{11}와 op_{12} 로 구성되었으며, 각각의 연산의 접근 모드는 w, w 로 설정되어 있다. 또한 객체 o_3의 범주 집합에는 o_{31}로 구성되었다. op_1와 o_3의 목적은 연산 op_{11}과 연산 op_{12}가 배타적 OR 관계를 유지하면서 단지 한번만 객체 o_{31}을 쓰기 연산을 수행한다.

$CAT(op_1) = \{(op_{11}, w), (op_{12}, w)\}$,

$CAT(o_3) = \{o_{31}\}$

$P(op_1, o_3) =$
$\langle\{(\widehat{op_{11}} \cdot \overline{\widehat{op_{12}}} + \overline{\widehat{op_{11}}} \cdot \widehat{op_{12}}, w), o_{31}\}\rangle$

[Ⅲ] 임무분리 무결성 보안제약조건

임무분리 무결성 보안제약조건은 연산 op_1의 범주 집합에는 op_{11}와 op_{12} 로 구성되었으며, 각각의 연산의 접근 모드는 w, w 로 설정되어 있다. 또한 객체 o_3의 범주 집합에는 o_{31}로 구성되었다. op_1와 o_3의 목적은 연산 op_{11}은 연산 op_{12}와 함께 병렬적으로 그리고 동기적으로 또는 비동기적으로 객체 o_{31}을 단지 한번만 쓰기 연산을 수행한다.

$CAT(op_1) = \{ (op_{11}, w), (op_{12}, w) \}$

$CAT(o_3) = \{ o_{31} \}$

$P(op_1, o_3) = \langle\{(\widehat{op_{11}}, w), o_{31}\} + \{(\widehat{op_{12}}, w), o_{31}\}\rangle$

[Ⅳ] 연결(concatenation)

두 개의 목적 P_v 와 P_w 를 연결하는 것은 P_v의 목적 뒤에 P_w 의 목적을 붙이는 연산이다. 즉, $P_v = \langle\{(op_i, w), o_j)\}\rangle$와 $P_w = \langle\{(op_{i+1}, w), o_j\}\rangle$라 하면, P_v 와 P_w 의 연결은 $P_vP_w = \langle\{(op_i, w), o_j\}, \{(op_{i+1}, w), o_j\}\rangle$이다.

[Ⅴ] 역(reverse)

어떤 목적의 역순은 주어진 목적들을 거꾸로 나열한 것이다. 즉, $(P_vP_w)^R = P_w^R P_v^R$ 이 성립한다. 만약, $P_vP_w = \langle\{(op_i, w), o_j\}, \{(op_{i+1}, w), o_j\}\rangle$이면, $(P_vP_w)^R = (P_w^R P_v^R)$
$= \langle\{(op_{i+1}, w), o_j\}^R, \{(op_i, w), o_j\}^R\rangle$
이 된다.

[Ⅵ] 접두사(prefix)와 접미사(surfix)

만약 $P_z = P_vP_w$ 라면 P_v 는 P_w의 접두사가 되고 P_w는 P_v의 접미사가 된다. 어떤 목적에서 접두사나 접미사를 제거함으로써 이루어지는 목적을 '서브 목적'이라 한다.

[Ⅶ] 길이

목적의 길이는 목적에 포함된 단위 목적의 개수를 말하는데 P_v와 같이 절대값을 써서 나타낸다. 만일 주어진 목적에 어떠한 단위 목적도 가지고 있지 않을 경우에는 '공 목적(empty purpose)'이라 하고 통상 λ 로 나타낸다. 따라서 $|\lambda| = 0$이고 모든 P_v에 대해 $\lambda P_v = P_v\lambda = P_v$ 가 성립한다.

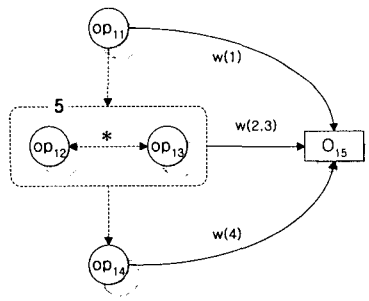
[Ⅷ] 반복

P_v 가 목적일 때 P_v^n이란 P_v를 n번 연결한 것이며 모든 P_v에 대해 $P_v^0 = \lambda$가 된다. 즉, $P_v = \langle\{(op_i, w), o_j)\}\rangle$ 라 하면,

$P_v^n = P_vP_v^{n-1} = \cdots = P_vP_v\cdots P_v^1$이 되므로,

$P_v^n = \langle\{(op_i, w), o_j)\}\rangle P_v^{n-1}$
$= \langle\{(op_i, w), o_j)\}, \cdots, \{(op_i, w), o_j)\}\rangle$ 이 된다.

한 객체에 대한 여러 객체들이 보안제약조건과 순서 및 반복에 의해 연속적인 접근 할당하는 예제에 대해 언급한다.



PBEAC 모델의 연산과 객체의 범주집합과 목적은 다음과 같다.

$CAT(op_1) = \{(op_{11}, w), (op_{12}, w), (op_{13}, w), (op_{14}, w)\}$,

$CAT(o_1) = \{o_{15}\}$,

$P_v = \{(\widehat{op_{12}}, w), o_{15}\} + \{(\widehat{op_{13}}, w), o_{15}\}$

$P(op_1.o_1) = \langle\{(op_{11}, w), o_{15}\}, P_v^5, \{(op_{14}, w), o_{15}\}\rangle$
$= \langle\{(op_{11}, w), o_{15}\}, (\{(\widehat{op_{12}}, w), o_{15}\} + \{(\widehat{op_{13}}, w), o_{15}\})^5, \{(op_{14}, w), o_{15}\}\rangle$

연산 op_1의 범주집합에는 연산 $op_{11}, \cdots, op_{14}$로 구성되고, 객체 o_1의 범주집합에는 o_{15}로 구성되었다. 먼저 서브 목적 P_v를 정의한다. 서브 목적 P_v은 연산 op_{12}와 연산 op_{13}이 함께 병렬적으로 그리고 동기적으로 또는 비동기적으로 객체 o_{15}을 단지 한 번만 쓰기 연산을 수행한다. 연산 op_1과 객체 o_1의 목적은 연산 op_{11}가 객체 o_{15}을 쓰기 연산을 수행하고, 서브 목적 P_v를 5번 반복한 다음에 연산 op_{14}은 객체 o_{15}을 쓰기 연산을 수행한다.

5. 결론 및 향후 연구방향

망 환경이 객체를 기반으로 하는 분산 컴퓨팅 환경으로 패러다임이 변화하고 있기 때문에 분산 객체를 기반으로 하는 접근제어 모델에 관한 연구가 요구된다. 본 논문에서는 분산 객체 기반의 객체들 사이에 발생하는 복잡한 보안 요구사항들을 보안규칙으로 제공함으로서 객체들의 무결성을 유지할 수 있도록 하고, 자율적 접근제어에 목적이라는 개념을 부가하여 특정 목적에 부합한 경우에만 접근이 가능하도록 강제적 접근제어 기능을 수행할 수 있는 목적지향 접근제어 기능을 제공한다. 또한 목적지향 접근제어의 불리언 표현과 연산을 정의하고, 이를 이용한 목적지향 BEAC 모델을 제안하였다.

향후 연구과제로는 목적지향 BEAC 모델을 확장하여 모빌 에이전트 보안에 응용할 수 있도록 연구를 수행할 예정이다.

참고문헌

[1] M. Yasuda, T. Tachikawa, and M. Takizawa, "Purpose-Oriented Access Control Model in Object-Based Systems" Proc. of the 2nd Australasian Conf. on Information Security and Privacy(ACISP'97), pp.38-49, 1997

[2] Randy Chow and I-Lung Kao "Modeling Complex Access Control Policies in Distributed Systems" IEEE 1995.

[3] Charles P. Pfleeger, Security in Computing, Prentice-Hall, 1989.

[4] David D. Clark and David R. Wilson, "A Comparison of Commercial and Military Computer Security Policies, "Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, April 1987, pp.184-194.

[5] S. Castano, M. Fugini, G. Matella, and P. Samarati, Database Security, Addison-Wesley, 1995.