논문 2008-5-26

상황인식 기반 적응적 접근제어 보안모델 설계에 관한 연구

A Study on Security Model Design of Adaptive Access Control based Context-Aware

김남일*, 김창복*

Nam-Il Kim, Chang-Bok Kim

요 약 본 논문은 기존의 접근제어 모델을 확장하여 상황인식 기반 접근제어 모델을 제안하였다. 본 논문에서 xoRBAC와 CAAC와 같은 상황인식기반 보안모델에 대한 최근연구들을 조사하였다. 정확한 정책평가를 위해 기존의 CAAC 보안모델에 상황브로커와 파인더 컴포넌트를 추가하였다. 이 보안모델에 의해 더욱 명확한 정책결정을 위해 상황정보 및 상황결정정보를 보다 용이하게 수집할 수 있다. 또한, 접근된 자원에서 또 한번의 사용자 이벤트를 판단하여, 접근할 수 있는 모든 가능한 자원들을 제어하였다. 본 논문에서 제안된 보안모델은 역할에 따른 특정정책과 제약조정을 통해 다양한 보안등급 및 접근권한방식을 동적으로 제공할 수 있다.

Abstract This paper is proposed context-aware based access control, model by extending original access control model. In this paper, we survey the recent researches about security model based context-aware such as xoRBAC and CAAC. For exactly policy evaluation, we make an addition Context Broker and Finder in existing CAAC security model. By this security model, Context information and context decision information is able to be collected easily for more correct policy decision. This paper controlled access of possible every resources that is able to access by user's event and constraint from primitive access resources. In this paper proposed security model can be offer dynamically various security level and access authority method alone with specified policy and constraint adjustment at user's role.

Key Words : 상황인식, 접근제어, GRBAC, CASA, 상황제한, 권한정책모델

I. 서 론

유비쿼터스(Ubiquitous)의 도래로 사용자의 위치, 시간과 관계없이 언제 어디서나 무의식 중에 정보가 처리되는 컴퓨팅 한경으로 변모하고 있다. 유비쿼터스 컴퓨팅 환경은 이동성과 다양한 네트워크가 특징이며, 주위상황의 변화가 매우 많으며, 주위 상황에 따라 동적이며, 상황 적응적으로 정보서비스를 제공해야할 필요성이 증가하고 있다. 상황인식(Context Aware)은 유비쿼터스 컴퓨팅과 더불어 지속적으로 연구 발전될 것이다.

최근, 상황인식을 이용한 보안 관련분야는 새롭게 주목을 받고 있으며, 상황인식을 이용하여 다양한 보안메카니즘을 적용하려는 시도가 활발히 진행되고 있다.[1]

기존의 보안모델은 사용자와 역할의 인증단계만을 거치고 자원에 대한 접근권한을 부여 하였다. 그러나 유비쿼터스 환경은 컴퓨팅 환경이 동적으로 변화하며, 변화되는 상황에 적응할 수 있는 보안메카니즘을 고려해야 한다. 따라서, 기존의 역할기반 접근제어 방식(Role Base Access Control)을 개선한 일반화 역할기반 접근제어(Generalized RBAC)방식과 상황제한 역할기반 접근제어(xoRBAC)방식 등 상황인식을 고려한 접근제어를 적용하려는 시도가 활발히 진행되고 있다.[2]-[4]

*정회원, 가천의과학대학교 IT학과
접수일자 2008.8.23, 수정완료 2008.9.23

본 논문에서는 기존에 상황인식을 통한 접근제어 보안모델에 대해서 조사하고, 기존의 접근제어 보안모델인 CAAC(Context-Aware Access Control)를 확장하여, 상황정보 및 타켓에 대한 속성정보를 체계적으로 수집할 수 있는 컴포넌트를 추가하였다. 즉, 파인더(Findr)와 상황브로커(Context Broker) 컴포넌트를 추가하여 정책결정에 필요한 모든 상황과 속성을 수집하도록 하였다. 또한, 본 논문에서는 사용자가 요구한 조건의 자원만을 접근하여 검색하는 것이 아니라 사용자의 특정 이벤트를 이용하여, 접근하고자 하는 자원과 관련된 모든 자원을 검색하는 방법을 제안함으로써, 유연한 접근 정책을 설계할 수 있도록 하였다. 본 논문에서 권한정책의 모든 상황제한조건과 이벤트는 실시간으로 관리자가 변경할 수 있게 함으로써, 다양한 보안등급 및 접근방식을 동적으로 제공할 수 있다.

II. 관련연구

1. 상황인식(Context-Aware)

상황(Context)은 "실세계에 존재하는 실체의 상태를 특징화하여 정의한 정보"라고 정의할 수 있다. 여기서 실체(Entity)란 인간, 장소 또는 인간과 서비스간의 상호작용을 의미한다. 실세계에서 상황이 무엇인지를 이해하기는 쉬우나 상황을 구체적으로 정형화하여 정의하는 것은 어렵다. 상황을 정형화하여 정의한 것을 상황정보(Context Information)라 하며, 사용자 환경에서 감정적인 상태, 주의력, 위치와 방향, 날짜와 시간, 사람과 사물 등 상황형태(Context Type)로 표현할 수 있다. 이와같이 정형화된 상황정보는 시스템 설계자가 이를 통해 다양한 어플리케이션에 응용할 수 있게 한다. 또한, 정보통합 및 재사용, 지식기반 구축 등 시스템 고도화의 관점에서 자원, 프로세스, 서비스의 계층화 및 세분화의 기준이 된다. 일반적인 상황정보는 다음과 같이 분류할 수 있다.

- 사용자 상황
- 물리적 환경 상황
- 컴퓨팅 시스템 상황
- 사용자-컴퓨터 상호 작용 이력
- 기타 미분류 상황

상황인식(Context Recognition)은 관련된 상황정보를 인지하고 상황지식(Context Knowledge)에 사상(mapping)하는 것이라 할 수 있다. 상황인식의 예로서, 위치서비스를 제공하는 네비게이션을 들 수 있다. 네비게이션은 GPS 위치서비스를 이용하여 운전자 주위상황에 따라 적절한 경로를 제공한다. 여기서 상황정보는 운전자 위치정보, 교통량정보, 신호등, 날씨조건 등이 될 수 있다. 네비게이션은 이러한 모든 상황을 추론해서 적응적인 경로 선택에 적용할 수 있다. 상황인식 시스템의 정의는 "사용자의 작업과 관련 있는 적절한 정보 또는 서비스를 사용자에게 제공하는 과정에서 상황을 사용하는 경우 이를 상황인식 시스템으로 정의"한다.[5]-[6]

2. 역할기반 접근제어 방식

역할기반 접근제어는 자원에 대한 사용자의 접근제어를 사용자의 역할 및 사용자가 속한 그룹의 역할에 의해 결정하는 특징을 가지고 있다. 역할기반 접근제어방식은 역할과 객체간의 관계로 접근권한을 관리함으로써, 사용자와 객체의 수가 대단히 많은 분산 기업환경에 적합한 특성을 제공한다.

Sandhu는 역할기반 접근제어를 다음과 같은 네 가지 모델로 구분하여 제안하였다.

- RBAC0 : 역할기반 접근제어를 기본모델
- RBAC1 : 기본모델에 역할의 상속개념인 역할 계층(Role Hierarchy)을 추가
- RABC2 : 기본모델에 상황제한(Context Constraint)조건을 추가
- RBAC3 : RBAC1과 RBAC2의 통합모델

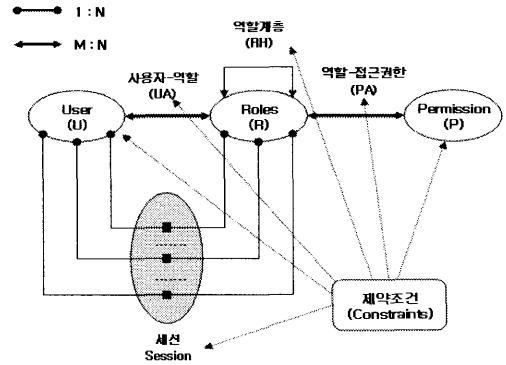


그림 1. 역할기반 접근제어
Fig. 1 Role Base Access Contorl

그림 2는 RBAC1과 RBAC2의 통합 모델인 RBAC3역할기반 접근제어 모델에 대해서 나타냈다.

① User(U) : 시스템의 정보를 사용하는 객체로서 한 사용자는 한 명의 사람에 대응된다.
② Role(R) : 접근제어 정책을 구현하는 중요한 의미적 구조로서, 조직내의 역할을 나타내며 고유의 권한과 의무를 갖는다. 한 사용자는 다수의 역할이 주어진다. 또한, 하나의 역할은 다수의 사용자와 관계가 있다.
③ Permission(P): 객체에 대한 특정 접근모드(읽기, 쓰기, 수정 등)의 승인을 나타낸다. 접근하고자 하는 객체는 역할하고 연결되어 진다.
④ Session(S): 시스템의 로그인을 통해 사용자가 수행하는 작업에 대한 역할의 활성화 상태이다. 이때 각 세션은 하나의 사용자와 여러 개의 권한을 매칭한다.
⑤ 사용자-역할&역할-접근권한 : 사용자-역할(UA) 관계는 사용자가 여러 역할을 가질 수 있고 동일 역할에 다수의 사용자가 할당될 수 있는 관계이며, 역할-접근권한(PA) 관계에서 역할은 다수의 접근권한이 부여될 수 있고, 동일 접근 권한에 여러 역할이 할당될 수 있는 관계이다.

기본적인 RBAC모델을 개선한 방법인 GRBAC는 기존의 모델에 상황정보를 추가한 방법으로, 접근제어 결정에 접근권한 정보를 주체(subject), 객체(object), 환경(environment), 연산(operation), 부호(sign) 등 5가지 튜플(tuple)로 표현함으로써, 접근제어 정책 기술의 단순함과 융통성을 제공한다. 여기서 주체는 역할로 표현되며, 객체는 접근하고자 하는 자원이다. 또한, 환경은 주체의 상황적 정보이며, 연산은 자원에 대한 접근 모드이다. 예를 들어 주치의의 역할을 할당 받은 사용자는 주말에 처방전을 읽을 수 없음을 다음과 같이 표현할 수 있다.

(<주치의, 처방전, 주말, 읽기>, −)

상황제한 역할기반 접근제어(xoRBAC)는 상황정보를 접근제어 결정에 이용하기 위하여 상황제한(Context Constraint) 조건을 사용하는 방법이다. 상황제한조건은 실시간 상황정보 속성의 실제 값과 접근제어 정책과 비교하여, 사용자 및 역할에 대한 권한부여를 하기 위해 사용된다. 상황제한 조건은 자원의 접근권한 결정을 위해 상황정보 속성이 충족되어야 할 조건을 기술한다. 상황제한 조건은 속성, 함수, 조건의 튜플(tuple)을 갖는다. 속성은 시간, 요일과 같은 동적으로 변화하는 속성을 나타내거나 위치, 소유관계, 생일, 국적과 같은 객체의 인스턴스에 따라 변화하는 속성을 나타낸다.[7]-[9]

Ⅲ. 기존 상황인식 시스템

CAAC(Context-Aware Access Control)는 상황제한을 이용한 접근제어 모델로서, 사용자의 접근요구에 대해 현재상황을 동적으로 측정하여 상황정보를 평가한다. 즉, 기존의 RBAC모델에 상황정보를 더하여 권한결정을 하는 모델이다. 본 모델의 접근정책은 3-튜플(User, Object, Action)을 기본으로 하여 상황제한이 추가된 모델이다. 본 모델의 접근제어 방법은 4가지 정의가 주어진다.

상황형태(CT:Context Type)는 상황정보를 정의하기 위해 상황제한의 요소로 사용한다. CT는 시간 또는 위치와 같은 일상 환경과 유사한 속성이 될 수 있으며, 인증 신뢰레벨과 같은 추상적인 개념일 수 있다.

상황제한(CC:Context Constraint)은 CT를 이용하여 상황정보를 정형화된 형식으로 정의한 것으로, 상황인식 보안설계에 있어서 복잡한 상황을 명세할 수 있어, 보다 세밀한 자원 접근제어가 가능하다. 상황제한은 다음과 같이 서술된다.

$$CC := CL1 \cup CL2 \cup \cdots \cup CLi \quad (1)$$

$$CL := CN1 \cap CN2 \cdots \cap CNi \quad (2)$$

$$CN := <CT><OP><VALUE> \quad (3)$$

만약 환자 데이터의 사용권한이 T(패스워드) 인증레벨로, am 8:00-pm 5:00사이에, 병원 안에서 인정되며, 시간과 위치상황이 TRUE가 아니면 더 높은 인증레벨이 요구된다고 가정하면, 상황제한은 다음과 같이 표현된다.

CC := (Time>=8:00 ∩ Time<17:00 ∩ Location in Hospital ∩ Trust=T(password)) ∪ (Trust>T(password))

권한정책(AP:Authorization Policy)은 사용자 또는 역할에 상황제한에 따라 자원의 접근권한을 제공하는 정책이다. 권한정책은 3-튜플로 나뉜다.

$$AP = (R, P, C) \quad (4)$$

R(Role)은 정책의 주체이며, 사용자의 역할을 의미한다. P(Permission)는 상황제한이 TRUE일 경우에 접근이 허용되는 자원과 접근모드로서 <M, O>로 표현한다. 여기서 M(Mode)는 자원에 대한 연산으로서, read, write, delete, update 등이 될 수 있다. 또한, O(Object)는 접근하고자 하는 자원이며, C(Constraint)는 상황제한이다. 여기서 상황제한이 비워있으면, 기본 역할기반 접근제어가 된다.

데이터 접근(Data Access : DA)은 사용자의 역할과 상황정보를 이용하여 특정 정보를 접근하고자 하는 시도이다.

$$Data\ Access = (U, P, RC) \quad (5)$$

U는 데이터 접근을 요구하는 사용자이며, P는 권한정책에서 표현한 자원에 대한 접근권한이며, RC(Runtime Context)는 CT의 실제 값을 나타낸다. RC는 다음과 같이 표현된다.

$$RC=\{v1\ of\ CT1, v2\ of\ CT2, \cdots\ vn\ of\ CTn\} \quad (6)$$

CAAC 접근제어 스키마는 인증엔진, 권한엔진 그리고 상황지식 저장소로 구분된다. 인증엔진(Authentication Engine)은 사용자 인증을 위한 것으로 사용자 ID를 권한엔진에게 제공한다. 권한엔진(Authorization Engine)은 사용자 ID에 해당하는 역할과 관련된 정책을 추출한다. 또한, 상황저장소에 정책평가에 관련된 상황과 DA에 의해 요구된 상황정보를 비교하여, 최종적인 평가를 한다. 그림 2에 CAAC 접근제어 스키마에 대해서 나타냈다.

CAAC모델은 헬스케어(Healthcare) 웹서비스에 적용하였으며, 의사와 환자들이 웹서비스 포탈을 통해 의료데이터를 접근할 수 있도록 하였다. 또한, 접근제어 정책명세 언어로 WS-Policy를 사용하였다.

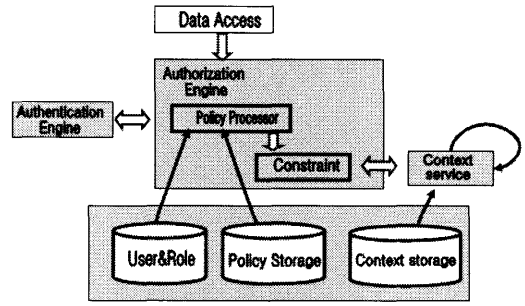


그림 2. CAAC 접근제어 스키마
Fig 2. CAAC Access Control Sch듬

Ⅳ. 접근제어 제안 모델

1. 상황인식 접근제어 보안모델

본 논문에서 설계한 접근제어 보안모델은 기존의 CAAC를 확장한 모델로서, 확장을 위해 다음과 같은 시스템 구조가 요구된다.

① 접근제어 보안시스템은 어플리케이션과 분리하여, 다양한 어플리케이션과 쉽게 연결할 수 있어야 한다.

② 정책평가 엔진은 사용자의 자원요구와 이에따른 모든 상황정보를 이용하여 보안정책을 평가할 수 있어야 한다. 정책 평가과정은 매칭방법으로 평가한다.

③ 상황정보와 주체, 객체, 행동, 환경 등 타겟에 대한 모든 속성정보를 수집할 수 있는 기능이 있어야 한다. 이를 통해 정책 평가엔진에서 상황에 따른 평가가 이루어진다.

④ 상황브로커(Context Broker)는 필요한 상황정보를 획득하는 컴포넌트이며, 미들웨어로 구성된다. 상황브로커는 센서, 상황정보제공자(CIP : Context Information Provider), 다른 시스템의 상황브로커로 부터 상황정보를 획득하며, 수집된 상황정보로부터 고수준의 상황정보로 추론한다.

⑤ 상황정보의 잘못된 수집에 대한 예외를 검출할 수 있어야 한다. 또한, 정책의 문법 및 데이터 에러를 처리할 수 있어야 하며, 결정할 수 없는 상황정보에 대해서는 디폴트 정책을 제공하여야 한다.

⑥ 접근제어 시스템에서 출력된 평가결과와 사용자 이벤트를 상황지식 모델에 입력하여, 권한이 부여된 자원 뿐아니라 사용자 이벤트에 관련된 모든 자원에 대한 접근을 제어하고자 한다.

본 모델은 이러한 시스템구조를 위해 다음과 같은 컴포넌트로 구성된다.

- PEP(Policy Enforcement Point) : 사용자의 자원에 대한 접근제어 요구와 평가결과 응답
- CKM(Context Knowledge model) : 사용자 이벤트와 관련하여 접근할 수 있는 모든 자원을 검색
- PDP(Policy Decision Point) : 입력된 요구에 대한 정책평가 및 평가결과를 PEP에 반환
- PAP(Policy Administrator Point) : 보안정책의 편집과 저장
- 파인더(Finder) : 보안정책 평가에 필요한 상황정보 및 정책결정정보(Policy Decision Information) 등을 수집
- 상황 브로커(Context Broker) : 상황정보 획득과 상황정보를 추론

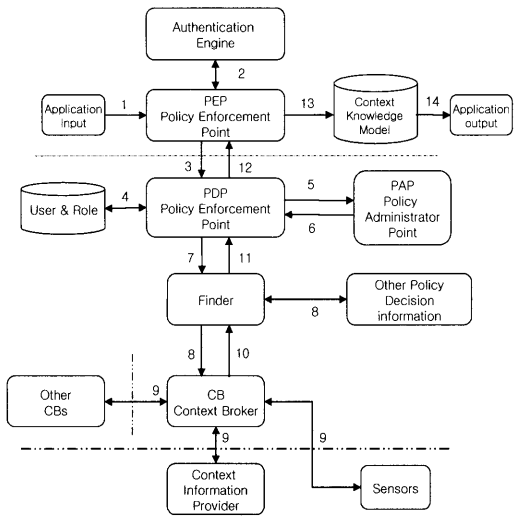


그림 3. 상황인식 보안 정책 모델
Fig 3. Context Aware Security Policy Model

PDP는 PEP와 접근요구 및 평가응답 메시지 전달, PAP로 부터 정책을 로딩, 정책에 사용되는 상황정보를 획득하기 위go 파인더와 통신할 수 있는 세 개의 인터페이스를 가지고 있다. 또한, PAP는 정책저장소에 저장되어 있는 정책들을 PDP에 제공하는 인터페이스가 있다. PAP는 정책저장소에는 상황정보를 이용할 수 없는 경우에 대비하여 일련의 디폴트정책이 제공된다. 파인더는 PDP에 상황정보전달, 상황브로커에 요청한 상황정보 쿼리(Query), 정책결정 정보 등과 통신할 수 있는 인터페이스가 있다.

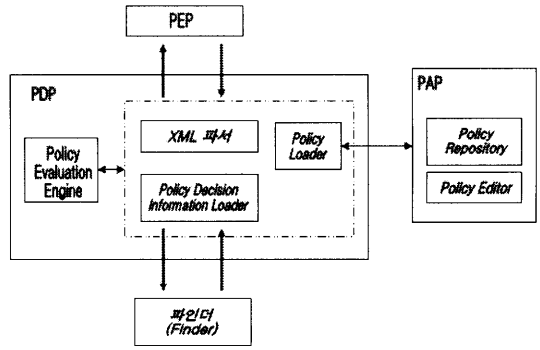


그림 4 . PDP와 PAP의 구성
Fig 4. PDP and PAP Configuration

상황브로커는 사용자의 입력요구에 대한 상황수집 및 수집된 상황정보에서 의미가 있는 고수준의 상황정보를 추론한다. 예를들어, 환자의 심장 박동수, 혈압 그리고 체온 등 센서 값에 따라 응급 상황을 추론할 수 있다. 상황브로커는 다른 장비의 상황브로커의 상황정보를 공유할 수 있다.

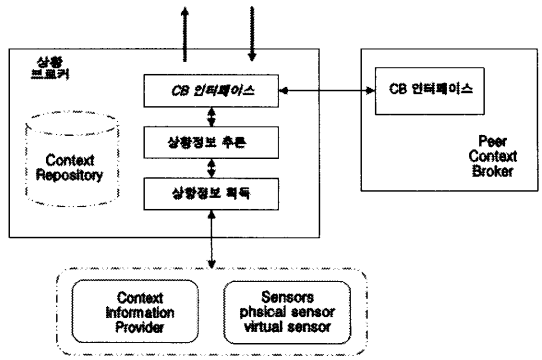


그림 5. 상황브로커의 구성 및 인터페이스
Fig 5. Context Broker and Interface

이와 같이 수집된 상황정보는 이후에 고수준의 상황분석을 위해 상황 레포지토리에 저장될 수 있다. 그러나 모바일 디바이스에서 상황정보 저장은 자원 한계의 이유로 선택적이다. 상황브로커는 센서, 상황정보 프로바이더(Context Information Provider), 파인더 , 다른 장비의 상황브로커 등의 인터페이스가 있다. 다음은 보안정책 모델에서 자원 접근에 대한 정책결정과정과 컴포넌트간 메시지 전달에 대한 순서에 대해서 나타냈다.

Step 1 : PEP는 어플리케이션으로부터 자원의 접근권한에 대한 요구를 입력을 받는다.

$$AReq := \{U_i, P_j, E_k\} \quad (7)$$

여기서 U_i는 사용자 인증정보이며, P_j는 접근하는 객체 또는 객체의 위치정보와 접근모드이다.또한, E_k는 사용자의 이벤트 또는 제한조건이다.

Step 2 : 인증엔진(Authentication Engine)을 통해 사용자를 인증한다. 사용자 인증은 사용자 ID나 지문, 홍체, 망막 등과 같은 바이오 정보일 수 있다.

Step 3 : PEP는 XML기반 접근제어언어인 XACML언어로 구성된 PDReq(PolicyDecisionRequest)메시지로 입력을 번역하여, PDP에 보낸다.

$$PDReq := \{ID_i, P_j\} \quad (8)$$

Step 4 : PDP에서 User & Role 저장소에 사용자 ID에 해당하는 역할을 검색한다.

Step 5 : PDP는 PAP에 정책을 요구하기 위해서 PReq(PolicyRequest)메시지를 보낸다. 여기서 R_n은 사용자의 역할이다.

$$PReq := \{R_n, P_j\} \quad (9)$$

Step 6 : PAP는 PDP에 XACML언어로 정책문장이 포함된 PRes(PolicyResponses)메시지를 보낸다. 여기서 C는 정책 저장소에 있는 상황제한이다.

$$PRes := \{R_n, P_j, C\} \quad (10)$$

Step 7 : PAP에서 전송받은 정책들로 부터 PDP는 어떤 상황정보와 정책 결정정보가 필요한지 확인하고, 파인더에 PDI(PolicyDecisionInformation)메시지를 통해 상황정보와 정책결정정보를 요구한다.

$$PDIReq := \{ID_i, Rn, Pj, C\} \quad (11)$$

Step 8 : 파인더는 상황정보를 수집하기 위해서 상황브로커에 CIReq(ContextInformation Request)메시지를 보낸다.

$$CIReq := \{ID_i, C\} \quad (12)$$

Step 9 : 상황브로커는 센서, CIP, 다른 상황브로커로부터 관련된 상황정보를 수집한다. 또한, 상황브로커는 PDP에서 편리하게 사용할 수 있도록, 수집된 상황정보를 추론한다.

Step 10 : 상황브로커는 실시간 상황정보와 상황추정보를 CIRes(ContextInformation Response)메시지를 파인더에 리턴한다. 여기서 RInf는 상황추론정보이다.

$$CIRes := \{v_1 \text{ of } CT_1, \cdots v_n \text{ of } CT_n\} \cup RInf \quad (13)$$

Step 11 : 파인더는 수집된 PDI(속성, 상황정보)메시지를 PDP에 보낸다. 또한, PDP는 파인더에서 전달된 상황정보와 정책결정정보들을 이용해서 보안정책을 평가한다. 여기서 TG_Att는 타겟에 대한 속성정보를 나타낸다.

$$PDI := \{v_1 \text{ of } CT_1, \cdots v_n \text{ of } CT_n\} \cup RInf \cup TG\_Att \quad (14)$$

Step 12 : PDP는 평가 결과를 XML언어 형식의 PDRes(PolicyDecisionResponse)메시지로 번역하여 PEP에 전송한다.

$$PDRes := Effect,\ Permit \text{ 또는 } Deny \quad (15)$$

Step 13 : PEP는 CKM에 RReq(Resource Request) 메시지를 보낸다. 이 메시지는 정책평가결과, 접근객체, 사용자 이벤트이다.

$$RReq := \{Effect, Pj, E_k\} \quad (16)$$

Step 14 : 정책평가 결과로 접근객체의 권한을 획득하며, 사용자 이벤트를 이용하여 사용자 접근객체를 기준으로 이벤트에 연결되어 있는 모든 객체를 검색하여 최종적으로 어플리케이션에 출력한다.

2. 상황지식 모델(Context Information Model)

본 논문에서 설계한 상항인식 접근제어에 있어서 사

용자가 요구한 조건의 자료만을 접근 가져오는 것이 아니라 사용자의 이벤트를 이용하여 접근할 수 있는 모든 자원을 검색하는 방법을 제안하였다. 예를들어, 주치의가 담당하고 있는 환자가 응급상황일 경우에, 응급상황이라는 이벤트에 필요한 자료 및 환자의 병력, 유사 응급상황에 관련된 처방법, 관련 전문의 등 응급상황에 관련된 모든 자료를 한번의 검색으로 검색하고 사용할 수 있도록 하는 것이다.

PDP에서 PEP에 전송된 자원접근 평가결과 메시지인 PDRes메시지가 'Deny'인 경우 CKM(Context Knowledge model)을 거치지 않으며, 'Permit'일 경우 사용자의 이벤트와 함께 CKM에 입력되어, 접근된 자원으로부터 이벤트에 관련된 모든 접근할 수 있는 자원을 접근할 수 있도록 한다. CKM에 입력되는 메시지는 다음과 같다.

$$RReq := \{Effect, P_j, E_k\} \quad (17)$$

여기서 P_j는 접근하는 객체 또는 객체의 위치정보이며, E_k는 접근된 객체부터 접근할 수 있는 모든 관련 객체들에 대한 이벤트 또는 제한조건이다. 본 논문에서 역할기반의 계층화된 보안과 주어진 역할이 만족되는 정책에서의 상황지식 서비스를 위해 다음과 같은 정형명세 모델을 정의한다.

$$Knowledge\_Model = \{P, P_0, T, L, C\} \quad (18)$$

① P는 보안정책에 대한 지식의 전체집합이다.
② $P_0 \subseteq P$는 최초 지식의 집합이다.
③ $R \subseteq P \times P$는 지식간의 전이를 나타내며 전체 관계로서, $\forall p \in P \cdot \exists p' \in P \cdot (p,p') \in T$. 이다. 즉, 모든 지식 $p \in P$에 있어 $(p,p') \in T$이면 하나의 상속자 $p' \in P$이 존재한다.
④ L은 보안정책 원소 명제의 유한집합
⑤ C는 정보에 접근할 수 있는 제한조건

$$image(P) = \{p' \mid \exists p \cdot \forall p \in P \wedge (p,p') \in R\} \quad (19)$$

상황 지식모델에서 모든 접근 가능한 상태는 초기상태 X_0로부터 고정점에 도달될 때까지 함수 image의 적용으로 계산된다. 다음은 접근가능한 모든 자원을 찾는 알고리즘에 대해서 나타냈다.

단계 1 : Function(p_n, e_m)
단계 2 : $P_0 := X_0$ repeatn
단계 3 : $P_{i+1} := P_i$ image(P_i) until ($P_{i+1} = P_i$)
단계 4 : if $P_{i+1} = P_i$ then stop
단계 5 : else 단계 3

[그림 6]는 상황지식모델에서 적응형 보안체계에서의 순차적 보안체계와 프로세스를 나타냈다.

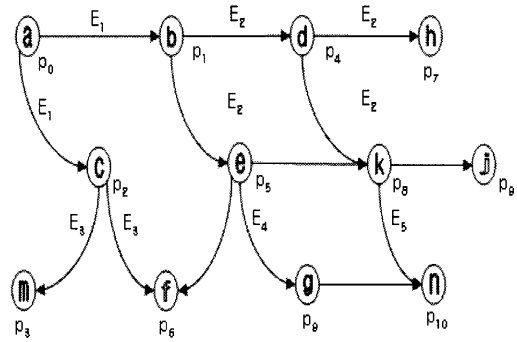


그림 6. 순차적 보안체계와 프로세서
Fig 6. Sequential Security System and Process

표 1. 상황지식 보안모델의 구조
Table 1. Context Knowledge Security Configuration

P	P_0	R	L	C
P_0	P_0	p_0p_1	a	E_1
		p_0p_2	a	E_1
P_1	-	p_1p_4	b	E_2
		p_1p_5	b	E_2
P_2	-	p_2p_3	c	E_3
		p_2p_6	c	E_3
P_4	-	p_4p_7	d	E_2
		p_4p_8	d	E_2
P_5	-	p_5p_6	e	-
		p_5p_8	e	-
		p_5p_9	e	E_4
P_8	-	p_8p_9	k	-
		p_8p_{10}	k	E_5

표 2. 제안모델과 기존모델의 장단점 비교분석

비교 내용	계층적 모델	CAAC 방식	제안된 모델
접근 방식	보안등급 기반 접근	역할기반 접근	역할 및 의미기반 접근
자원접근단위	개체단위	개체단위	개체 및 의미가 연결된 정보
상황정보	적용안함	부분적용	상황정보 확장적용
정책수정	전반적 수정 필요	부분 수정 필요	실시간 동적수정
정책추가	보안체계 전체 수정을 통한 정책 추가	보안체계 부분 수정을 통한 추가	기존 보안체계 수정없이 추가
권한관리	평면적인 단위의 관리	동적인 권한 관리	동적인 권한 관리

만약, App := {Permit, p_1, E_2} 메시지가 PEP로부터 CKM에 전송된다, 접근 가능한 모든 공간 집합은 다음과 같이 계산된다.

Function(p_1, E_2) : init = {p_1}
repeat1 = {p_1, p_4, p_5}
repeat2 = {p_1, p_4, p_5, p_7, p_8}
repeat3 = {p_1, p_4, p_5, p_7, p_8, p_9}
repeat4 = {p_1, p_4, p_5, p_7, p_8, p_9}
if repeat3 = repeat4 : Fixed Point

여기서 접근 가능한 자원으로의 전이는 동일한 이벤트 또는 제한조건이 같거나, 이벤트가 없는 경우에 가능하다. 따라서 ARes := {permit, p_1, E_2}를 만족하는 정책 집합에서의 원소는 다음과 같다.

Application Output = { b, d, e, h, k, j } (20)

기존의 자원에 대한 접근제어 모델은 하나의 정보에 대해서 권한을 부여하여, 상황에 따른 정책변경을 개별적으로 제어하여야만 했다. 그러나 본 논문에서 설계한 상황인식 접근제어에 있어서 사용자가 요구한 조건의 자원만을 접근하는 것이 아니라 사용자의 이벤트나 제한상황을 이용하여 접근이 허용된 객체 그리고 이벤트에 관련된 모든를 접근할 수 있는 방법을 제안하였다. 본 논문에서 설명된 지식기반의 보안모델에서는 역할에 따른 특정정책과 제약의 조정을 통해 다양한 보안등급 및 접근방식을 제공할 수 있다.

V. 결론

본 논문에서는 기존에 상황인식을 통한 접근제어 방식에 대해서 조사하고, 상황인식을 이용한 접근제어 시스템을 설계하기 위해, 요구사항을 기술하였다. 또한, 요구사항을 만족하기 위해서 기존의CAAC(Context Aware Access Control)모델을 확장

하여, 상항정보 및 타켓에 대한 속성정보를 체계적으로 수집할 수 있는 컴포넌트를 추가하였다. 이를 위해 파인터와 상황브로커 컴포넌트를 추가하여 구성하였다. 특히, 본 모델을 접근제어 시스템에서 출력된 평가결과와 사용자 이벤트를 상황지식 모델에 입력하여, 권한이 부여된 자원 뿐 아니라 사용자 이벤트에 관련된 모든 자원에 대한 접근을 제어하고자 하였다. 이 방법을 통해 접근하고자 하는 정보 뿐 아니라 특정 이벤트 또는 조건에 맞는 모든 정보를 탐색할 수 있어, 유연한 접근 정책을 설계가 가능하게 되었다. 또한, 권한 정책의 모든 제한조건과 이벤트는 실시간으로 관리자가 변경할 수 있게 함으로써, 다양한 보안등급 및 접근방식을 동적으로 제공할 수 있다. 본 논문에서 상황인식 기반 접근제어 방식은 보안 메카니즘에서 자원의 접근제어 및 개인 프라이버시 정보보호를 위해 많은 장점을 갖는다는 것을 알 수 있었다. 앞으로의 연구진행 방향은 본 논문에서 제안한 접근제어 보안모델을 접근제어 정책언어로 구현하고자 한다.

참고문헌

[1] W.Edwards, "Discovery systems in Ubiquitous Computing", IEEE Pervasive Computing, pp.70-77, April-june 2006

[2] Biil Schilit, Norman Adams, and Roy Want, "Context-Aware Computing Applications," Proceedings of the Workshop on Mobile Computing System and Applications, pp. 85-90, 1994.

[3] Harry Chen, Tim Finin, and Anupam Joshi, "An Intelligent Broker for Context-Aware Systems," Adjunct Proceeding of Ubicomp 2003, pp. 12-15, Oct. 2003.

[4] Anind K, Daniel Salber, and Gregory D. Abowd, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," Anchor article of a special issue on Context-Aware Computing in the Human-Computer Interaction(HCI) Journal, Vol.16(2-4), pp. 97-166, 2001.

[5] Ravi Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E. Youman, "Role-Based Access Control Models," IEEE Computer, Volume 29, Number 2, pp.39-47, 1996.

[6] Guangsen Zhang and Manish Parashar, "Dynamic Context-aware Access Control for Grid Application," The Fourth International Workshop on Grid computing 2003, pp. 101-108, 2003.

[7] Michael J. Covington, Wende Long and Srividhya Srinivasan, "Secure Context-Aware Applications Using Environment Roles," Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies, May 2001, Chantilly, Virginia, USA.

[8] M. J. Moyer and M. Abamad, "Generalized Role-Based Access Control," 21st International Conference on Distributed Computing Systems, April 16-19, 2001, Atlanta, GA, USA.

[9] Marc Wilikens, Simone Feriti, Alberto Sanna and Marcelo Masera, "A Context-Related Authorization and Access Control Method Based on RBAC: A case study from the health care domain," Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, June 2002, Monterey, California, USA.

저자 소개

김 남 일(정회원)

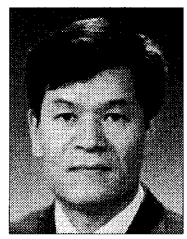
·1989년2월 : 단국대학교 전자공학과 석사

·2008년 10월현재 : 인천대학교 컴퓨터공학과 박사수료

·2008년10월 현재 : 가천의과학대학교 IT학과 교수

<관심분야> 이동통신, 인터넷 보안, 임베디드시스템

김 창 복(정회원)



·2000년8월 : 건국대학교 전자공학과 박사

·2008년10월 현재 : 가천의과학대학교 IT학과 교수

<관심분야> 컴퓨터네트워크,트래픽 제어, 유비쿼터스,유헬스케어,BcN