SMV를 이용한 접근통제모델의 정형적 설계방법 연구[1)]

황대연[O*] 강인혜** 강필용*** 이완석*** 최진영*
*고려대학교 컴퓨터학과
{dyhwang[O], choi}@formal.korea.ac.kr

**서울시립대학교 기계정보공학과
inhye@uos.ac.kr

***한국정보보호진흥원
{kangpy, wsyi}@kisa.or.kr

A Study on Formal Design of Access Control Model using SMV

Dae-Yon Hwang[O*] Jin-Young Choi*
*Dept. of Computer Science & Engineering, Korea University

In-Hye Kang**
**Dept. of Mechanical and Information Engineering, University of Seoul

Pil-Yong Kang***, Wan S. Lee***
***Korea Information Security Agency

요 약

컴퓨터 시스템에 대한 보안의 필요성이 계속적으로 증대되고 있으며 이에 다양한 보안시스템들이 개발되고 있다. 이러한 보안 시스템들이 높은 등급의 평가를 받기위해서는 정형적 방법론을 사용하여 명세 및 검증을 해야 한다. 본 논문에서는 정형 검증의 한 방법론인 모델 체킹을 이용하여 접근통제모델을 설계하고 검증하는 방법을 제안하고자 한다.

1. 서 론

컴퓨터 시스템이 발달하고 사회에서 핵심적인 역할을 하게 되고 있는 가운데 그와 비례하여 컴퓨터 시스템이 더욱 많은 보안 위협에 노출되고 있다. 이러한 위협으로부터 컴퓨터 시스템들을 안전하게 보호하고자 하는 연구가 전 세계적으로 활발하게 진행되고 있다. 이러한 의도를 가지고 만들어진 보안 제품의 품질 보증을 하기 위해 국가별로 평가 제도를 시행하고 있다. 국내에서는 정보보호진흥원에서 보안 제품에 대한 품질을 평가하고 등급을 부여하고 있다[1]. 등급은 K1에서 K7까지 총 7단계로 나누어지는데, 5단계 이상이 고등급의 제품이며, 고등급 판정을 받기 위해서는 정형적인 방법론을 제품에 적용해야한다. 최근에는 국제 공통의 평가기준인 CC(Common Criteria)[2]를 제정하여 서로 다른 나라에서 개발한 보안 제품에 대해서도 공통된 기준으로 평가를 수행하고 있다. 그리고 이 CC에서도 마찬가지로 고등급의 평가를 받기 위해서는 설계단계에서부터 정형적인 방법론의 적용이 필요하다. 하지만 이러한 정형적인 방법론을 보안 시스템에 적용하는 것이 쉽지 않아 아직까지도 보안 시스템을 명세하고 그 안전성을 검증한 사례는 많지 않다.

1) 본 연구는 한국정보보호진흥원 위탁과제로 수행되었음

본 논문에서는 정형 검증 도구의 하나인 SMV를 이용하여 보안 시스템의 접근통제모델을 설계하고 검증할 수 있는 방법을 제시하고자 한다.

본 논문은 다음과 같이 구성되어있다. 2장은 접근통제에 대해 설명하고 3장에서는 보안 모델을 유한상태기계로 어떻게 표현하는지를 설명하며, 4장에서는 만들어진 유한상태기계를 기반으로 간단한 모델을 SMV를 이용하여 모델링 하였다. 5장은 모델 체킹하여 안전성을 검증한 예제를 설명한다. 마지막으로 6장에서는 결론 및 향후 연구방향을 제시하고자 한다.

2 접근통제

시스템에서 안전성을 보장한다는 것은 시스템의 자원들에 대해, 접근이 허용되지 않은 프로세스의 자원에 대한 접근을 막는 것이다. 시스템의 자원에 대한 프로세스의 접근을 인증하는 것을 접근통제라 하며, 시스템의 안전성은 접근통제를 실행하는 보안 도구의 능력에 의존된다. 접근통제는 합법적인 개체의 활동을 제한하는데 관여하고, 접근 통제는 사용자가 객체로 접근을 시도하는 모든 것을 조정하는 참조 모니터에 의하여 시행된다. 이러한 모니터링은 프로세스가 자원에 접근할 수 있는 권한이 있는지를 인증 데이터베이스를 참조하면서 확인한다. 이 인증 데이터베이스는 보안 관리자에 의해 관리되

고, 관리자는 보안 정책을 근거로 이러한 인증들을 설정한다.

3. 유한상태기계 기반의 접근통제모델

시스템의 접근통제를 보여주는 접근통제모델은 여러 가지의 방법으로 표현이 가능하나, 본 논문에서는 유한상태기계로 표현하였다.

● 시스템

시스템 Σ는 상태기계로 다음과 같다.

$$\Sigma = \{S^{\Sigma}, T, s_{init}^{\Sigma}, Q\}$$

S^{Σ} : 시스템 상태들의 집합.

Q : 상태의 전이를 일으키는 요청들.

T : 상태 전이 함수. $T: Q \times S^{\Sigma} \rightarrow S^{\Sigma}$. s_i^{Σ}에서 요청 q를 받으면 상태 s_{i+1}^{Σ}= T(q,s_i^{Σ})로 전이.

s_{init}^{Σ} : 시스템의 초기 상태.

● 보안 모델

보안 모델[3] M은 다음과 같은 집합이다.

$$M = \{S, R, C\}$$

S : 모델에 의해 정의된 시스템 보안 상태들의 집합

R : 접근 통제 규칙들의 집합.
r(s,s')의 형식을 가진다. s에서 s'로의 전이가 접근통제 모델에 부합되는지를 검사한다.

C : 상태 보안 기준들의 집합. c(s)의 형식으로 표현되고 상태 s의 보안을 검사한다.

상태 s∈S는 안전하다는 것은 다음을 만족한다는 이야기이다. $\forall c \in C: c(s) = true$

● 시스템 안전성 (system safety property)

시스템 안전성은 다음과 같이 표현할 수 있다.

$$\Lambda = \{M, \Sigma, D\}$$

M : 보안 모델, $M = \{S, R, C\}$

Σ : 시스템 $\Sigma = \{S^{\Sigma}, T, s_{init}^{\Sigma}, Q\}$

D : 시스템의 상태들과 모델의 시스템 보안 상태들의 관계를 설정하는 함수, $D: S^{\Sigma} \rightarrow S$

시스템 안정성을 이용하여 시스템의 안전성 정리를 다음과 같이 표현할 수 있다.

보안 모델 M을 구현한 시스템 Σ은 다음 조건을 만족하면 안전하다.

$\forall c \in C: c(D(S_{init}^{\Sigma})) = true$

$\forall s_i^{\Sigma}, s_{i+1}^{\Sigma} \in S^{\Sigma}$: s_{i+1}^{Σ}=T(q,s_i^{Σ}) $\exists s_i$ = D(s_i^{Σ}), s_{i+1} = D(s_{i+1}^{Σ}) 이면 $\forall r \in R$: r(s_i, s_{i+1}) = true

$\forall s_i^{\Sigma} \in S^{\Sigma}: s_{init}^{\Sigma}$ 에서 상태 s_i^{Σ} 에 도달 가능하면, s_i = D(s_i^{Σ}): ∀c∈C: C(s_i) = true

4. 접근통제모델의 검사

4.1 모델 체킹과 SMV

모델 체킹은 유한상태 시스템이 원하는 속성을 만족하는가를 자동으로 분석하는 정형 검증 기술이다. 즉, 상태 전이 시스템과 속성이 주어지면, 알고리즘에 의해 주어진 시스템이 검증하고자 하는 속성을 만족하는지를 파악하기 위해 전체 상태 공간을 검사한다.

SMV는 SMV 입력 언어로 모델링된 시스템이 CTL 요구 명세로 표현한 보안 기준을 만족하는지를 자동적으로 검증하는 정형 검증 도구이다. SMV 입력 언어는 시스템을 동기적인 밀리 머신이나 비동기적인 네트워크로 쉽게 명세할 수 있는 특성을 가지고 있다[4].

4.2 SMV을 이용한 접근통제 모델링 예제

[표 1]은 본 논문에서 대상으로 하는 접근통제 리스트이다.

	High object	Low object
High subject	읽기,쓰기	읽기
Low subject		읽기, 쓰기

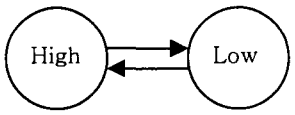
[표 1] 접근통제 리스트

[표 1]에서 보듯이 시스템에는 두 그룹의 주체가 있고 두 그룹의 객체가 있다. 높은 보안 등급을 갖는 주체는 높은 보안의 객체에 읽고, 쓰는 것이 가능하지만, 낮은 보안의 객체에는 쓰기 권한은 받을 수 없다. 반대로 낮은 보안 등급의 주체는 높은 보안 등급을 가지는 객체를 읽거나 쓸 수 없고, 낮은 보안의 객체에는 읽기, 쓰기가 모두 가능하다. 즉 no read up, no write down의 규칙을 가지고 있는 접근통제모델을 나타낸다.

4.2 접근통제모델

접근통제모델은 주체에 대한 부분과 객체에 대한 부분으로 나누어진다. 각 주체는 높은 등급인 high상태나 낮은 등급의 low 상태를 가질 수 있다. 주체의 등급을 변경하는 요청이 들어오면 주체의 상태가 전이된다.

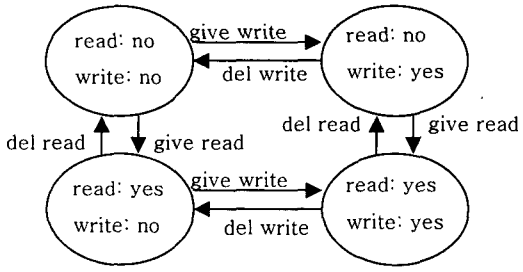
[그림 1]은 주체가 가질 수 있는 상태를 표현한 것이다.



[그림 1] 주체의 등급 상태 전이도

각 객체들은 주체들에 대한 접근통제목록을 기준으로 상태를 정의했다.

[그림 2]는 하나의 객체가 하나의 주체에 대해 가질 수 있는 상태들을 표현한 것이다. 하나의 객체가 한 주체에 대해 가질 수 있는 상태는 4가지이며 상태간의 전이는 주체에게 접근 권한을 주거나 접근 권한을 삭제하는 요청에 의해 일어난다. 각 객체의 모든 주체에 대한 현재 상태를 모은 것이 바로 객체에 대한 주체의 접근통제목록이 된다.



[그림 2] 객체의 접근통제목록 상태도

전체 접근통제모델의 상태기계는 모든 주체의 상태기계와 각 객체들의 모든 주체들에 대한 접근통제목록 상태기계의 카티시안 곱이다. 시스템의 상태는 현재 객체들의 접근통제목록과 주체들의 상태 속성이 된다. 시스템의 전이는 권한의 부여, 권한의 삭제 또는 주체의 등급 변화 중 하나의 요청에 의해 일어난다.

접근통제모델에서 전이가 일어날 때 검사하는 접근통제규칙은 다음과 같다.

- 낮은 등급의 객체의 권한을 변경하는 경우 높은 주체에게 쓰기 권한을 주지 못한다.
- 높은 보안의 객체의 권한을 변경하는 경우 낮은 주체에게는 어떠한 권한도 주지 못한다.

4.3 보안 기준

다음은 접근통제모델이 만족해야할 보안 기준들이다.

보안 기준 1 : 항상 모든 높은 등급의 주체는 낮은 등급의 객체에 대해 쓰기 권한을 갖지 못한다.

보안 기준 2 : 항상 모든 낮은 등급의 주체는 높은 등급의 객체에 대한 어떠한 권한도 갖지 못한다.

이 보안 기준들을 CTL을 이용하여 SMV에 넣을 속성으로 만들면 각각 다음과 같다.

속성 1: SPEC AG !(obj[2].sub[subject].p[2] = 1
& sub[subject] = high)

속성 2: SPEC AG !((obj[1].sub[subject].p[1] = 1
| obj[1].sub[subject].p[2] = 1)
& sub[subject] = low)

이 모델에서는 2개의 주체와 2개의 객체를 만들었다. 객체 obj[1]은 높은 보안 등급을, 객체 obj[2]는 낮은 보안 등급을 갖도록 하였고, 주체 sub[1]은 초기 상태를 높은 등급으로, 주체 sub[2]는 초기 상태를 낮은 등급으로 설정하였다.

5 SMV를 이용한 분석 결과

구현한 접근통제모델을 SMV를 이용하여 모델 체킹을 하면 원하는 보안 기준을 만족하는지를 보여준다. 보안 속성을 만족하지 않는다는 것은 구현한 접근통제모델이 안전하지 않다는 것이며 SMV 만족하지 않는 보안 속성에 대한 반례를 보여준다.

예제 모델의 경우 두 가지 속성에 대해 모두 안전하지 않다는 모델 체킹 결과가 나오며, 다음의 [그림 3]은 첫 번째 보안 기준에 대한 반례이다.

	1	2	3	
action	4	3	-	
grantPerOk	1	0	-	
obj[2].sub[1].p[2]	0	0	0	
obj[2].sub[2].p[2]	0	1	1	
object	2	1	-	
pgt	2	1	-	
prohibitPerOk	0	0	-	
sub[1]	high	high	high	
sub[2]	low	low	high	
subject	1	2	2	
tarsubject	2	1	-	

[그림 3] 속성 1에 대한 반례

반례를 살펴보면 주체 sub[2]가 낮은 등급인 상태에서 객체 obj[2]에 대한 쓰기 권한 obj[2].sub[2].p[2]를 습득한다. 다음 상태에서 주체 상태 요청이 일어나 주체 sub[2]의 등급이 높은 등급으로 변하여 보안 기준에 위배되는 상태에 도달했음을 보여준다.

6. 결론 및 향후 연구 방향

접근통제모델의 안전성을 정형적으로 명세하고 검증한 사례는 많지 않다. 본 연구에서는 일반적으로 널리 알려져 있는 정형 검증 도구인 SMV를 이용하여 접근통제모델의 안전성을 명세하고 분석해 보았다.

이러한 방법론을 적용함으로써 보안시스템을 개발하기 전에 설계단계에서부터 보안 모델의 안전성을 분석하고, 개발 단계에서 발생할 수 있는 오류를 줄일 수가 있다.

향후에는 보다 효율적인 구현을 통해 보다 실제 세계의 시스템에 가까운 모델을 명세하고 검증하고자 한다. 또, 일반 개발자들이 사용하기에는 복잡한 SMV 입력언어 대신 보다 쉽게 접근할 수 있는 방법을 개발하고자 한다.

7. 참고문헌

[1] 정보보호시스템 평가·인증 가이드, 한국정보보호진흥원, 2002. 12.

[2] Common Criteria for Information Technology Security Evaluation Version 2.1, August 1999.

[3] 동적인 보안분석 평가도구 개발, 한국정보보호진흥원, 2002. 12.

[4] E. M. clarke, O. Grumberg, ad D. A. Peled, "Model checking", MIT Press, 1999.