# 보안모델 및 정형검증 도구 개발[1)]

김일곤[0*], 최진영*, 강인혜**, 강필용***, 이완석***, Dmitry P. Zegzhda****

*고려대학교 컴퓨터학과
{igkim[0], choi}@formal.korea.ac.kr,

**서울시립대학교 기계정보공학과
inhye@uos.ac.kr

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

**** St.-Petersburg State Polytechnical University
dmitry@ssl.stu.neva.ru

# Development of Security Model Verification Tool

Il-Gon Kim[0*], 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

Dmitry P. Zegzhda****
**** St.-Petersburg State Polytechnical University

요약

보안 시스템에 대해서 고등급 평가를 받기 위해서는 정형적 방법론을 사용하여, 보안 모델을 설계하고, 보안 속성을 정확히 기술해야만 한다. 본 논문에서는 정형적 설계 방법을 통해 보안모델을 설계하고 검증하기 위한, SPR(Safety Problem Resolver) 정형검증도구의 검증방법 및 기능에 대해 소개하고자 한다.

## 1. 서론

정보통신기술의 발달과 더불어 정보시스템에 대한 의존도와 활용도가 증가되고 있는 반면에, 그에 대한 역기능으로 각종 보안 위협에 쉽게 노출되어 있는 실정이다. 이에 따라 사용자 개인뿐만 아니라 국가 기밀정보를 외부의 악의적인 공격자로부터 보호하기 위해, 보안 운영체제, 침입탐지 시스템, 방화벽등과 같은 보안 시스템을 개발하기 위한 연구가 한창 진행중에 있다. 국내의 경우, 매년 보안시장의 규모가 점차 커져가고 있다. 이런 보안제품을 국외에 수출하기 위해서는 보안평가 체계에 따른 등급심사 과정을 거쳐야만 한다. 국내의 경우 K1부터 K7까지 등급을 분류하고 있으며, K5 이상의 고등급 평가를 받기 위해서는 기능, 기본 상계 설계시 정형적 방법론을 이용해야만 한다[1]. 국제공통평가 기준인 CC(Common Criteria)[2]의 경우에도 EAL1부터 EAL7까지 등급을 분류하고 있으며, EAL5 이상의 고등급을 받기 위해서는 정형적 방법론을 사용해야만 한다. 하지만 아직까지 정형적 방법론을 사용하여 보안모델[3][4]을 설계하고 안전성을 검증한 사례는 거의 존재하지 않는다. 따라서, 본 논문에서는 보안모델을 명세하고 검증할 수 있는 SPR(Safety Problem Resolver) 정형검증도구의 검증방법 및 기능을 소개하고자 한다.

본 논문의 나머지 부분은 다음과 같이 구성되어 있다. 제2, 3장에서는 보안모델의 안전성을 분석하기 위한 SEW(Security Evaluation Workshop) 구조와 SPR 도구에 대해 간략히 소개하고, 제4장에서는 간단한 접근통제모델[5]의 안전성을 SPSL로 명세하고 분석한 예제에 대해 설명하고 마지막으로 제5장에서는 결론 및 향후 연구방향을 제시하고자 한다.

[1)] 본 연구는 한국정보보호진흥원 위탁과제로 수행되었음

## 2. SEW(Safety Evaluation Workshop)

이 논문에서 언급하고 있는 보안시스템이란 운영시스템, 침입탐지시스템, 방화벽등을 그 대상으로 하고 있다. 보안시스템의 안전성을 분석하기 위해서, SEW라고 불리는 평가구조 체계를 사용하고자 한다. SEW을 구성하고 있는 중요 컴포넌트들은 [그림 1]에 잘 나타나 있다.

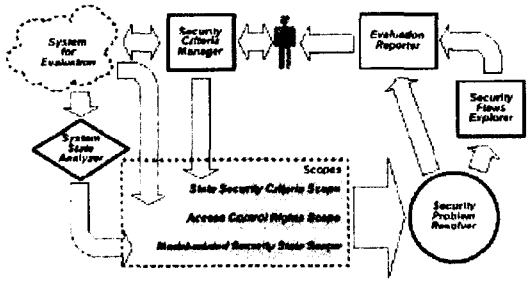


그림 1. SEW(Safety Evaluation Workshop) 구조

SEW를 구성하고 있는 각각의 중요 컴포넌트들의 기능을 살펴보면 다음과 같다.

1. System State Analyzer : 보안모델의 상태를 자동추출
2. Security Criteria Manager : 보안모델이 만족해야 하는 해당 보안속성을 GUI 형태로 입력
3. Scopes :
   1) State Security Criteria Scope : 보안속성(예, No Read Up, No Write Down등)
   2) Access Control Rights Scope : 접근통제권한(예, Security Reference Monitor[6]등)
   3) Model-related System Security Scope : 보안모델 상태(예, 주체(subject), 객체(object), ACL등)
4. SPR : 프롤로그(Prolog)언어 기반의 정형검증도구, SWI-Prolog[7]로 구현
5. SPSL(Safety Problem Specification Language) : 3번에서 언급한 3개의 Scopes를 나타내기 위한 프롤로그 기반의 명세언어.
6. Security Flows Explorer : SPR 도구를 통한 보안취약점 추적(프롤로그 기반의 역추적 결과들)
7. Evaluation Reporter : 보안취약점에 대한 최종 보고서

## 3. SPR(Safety Problem Resolver) 도구

SPR은 보안모델의 안전성을 분석하기 위한 정형검증도구로서, 프롤로그 언어를 기반으로 하고 있으며, 3개의 Scopes를 입력파일로 받아들인다. SPR 도구의 API는 C++언어로 작성되어 있다. 3개의 Scopes 입력파일들은 보안 모델을 나타내게 된다. 즉, SPR 도구에서 입력으로 받아들이는 보안모델은 3개의 컴포넌트로 설명할 수 있다.

보안모델 = 시스템 보안상태 + 접근통제규칙 + 보안기준

## 4. 접근통제모델 분석 예제

앞에서 언급하였듯이, SPR 도구를 이용하여 보안시스템의 안전성을 분석하기 위해서는 3개의 Scopes로 구성되어 있는 보안모델을 입력파일로 받아들여야 한다. 제4장에서는 간단한 접근통제모델을 어떻게 SPSL로 명세하고 SPR 도구를 통해 해당 보안속성을 검증하는지 살펴보도록 하겠다. [그림 2]는 3개의 주체와 3개의 객체로 구성된 간단한 접근통제모델을 보여주고 있다.

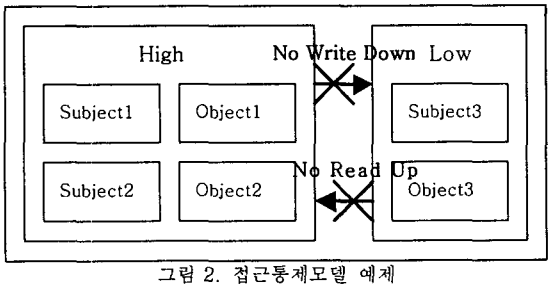


그림 2. 접근통제모델 예제

위의 그림에서 보여주고 있는 접근통제모델은 상위(High)와 하위(Low) 그룹으로 나뉘어져 있으며, 상위그룹에는 Subject1, Subject2, Object1, Object2가 속해있고, 하위그룹에는 Subject3와 Object3가 속해있다. 위 예제에서 Subject는 사용자를 나타내며, Object는 사용자가 소유하고 있는 파일들을 가리키고 있다. 그리고 상위 모델의 경우 다음과 같은 보안속성을 보장해야만 한다.

1) No Read Up: 하위그룹은 상위그룹의 객체를 읽을 수 없다.
2) No Write Down: 상위그룹은 하위그룹의 객체에 쓸 수 없다.

[표 1]은 위에서 언급한 접근통제모델에서 주체와 객체사이의 권한에 대한 접근통제 리스트를 보여주고 있다.

표 1. 접근통제 리스트 예제

| 주체 \ 객체 | Object1 | Object2 | Object3 |
|---|---|---|---|
| High Group | 읽기<br>쓰기 | 읽기<br>쓰기 | 읽기 |
| Low Group | | | 읽기 |
| Subject1 | 읽기<br>쓰기 | 읽기<br>쓰기 | 읽기<br>쓰기 |
| Subejct2 | 읽기<br>쓰기 | 읽기<br>쓰기 | 읽기 |
| Subject3 | | | 읽기<br>쓰기 |

제3장에서 언급하였듯이, 보안모델의 안전성을 평가하고 분석하기 위해서는 3가지의 입력파일을 SPSL로 명세한 후, SPR 도구를 통해 그 취약점을 확인하게 된다. 3개의 입력파일은 각각 다음과 같은 .sc라는 파일 확장자명을 가져야 한다. 위의 예제에 대한 시스템 보안상태는 [그림 3]과 같이 표현 될 수 있다.

```
subjectAttr(subjectGroups).
subject(s1,[subjectGroups(high)]).
subject(s2,[subjectGroups(high)]).
subject(s3,[subjectGroups(low)]).
objectAttr(objectType).
objectAttr(high).
objectAttr(low).
objectAttr(s1).
objectAttr(s2).
objectAttr(s3).
object(o1, [objectType(file), high(rd,rp,wd,wp), low,
s1(rd,rp,wd,wp), s2(rp,rd,wd,wp), s3]).
object(o2, [objectType(file), high(rd,rp,wd,wp), low,
s1(rd,rp,wd,wp), s2(rd,rp,wd,wp), s3]).
object(o3, [objectType(file), high(rd,rp), low(rd,rp),
s1(rd,rp,wd,wp), s2(rd,rp), s3(rd,rp,wd,wp)]).
```

그림 3. 시스템 보안상태 예제

[그림 3]은 [그림 2]에서 언급한 접근통제모델에 대한 시스템 보안상태를 SPSL로 기술한 부분을 나타내고 있다. 이 접근통제모델이 No Read Up, No Write Down의 보안속성을 만족시키는지 체킹하기 위해, [그림 4]와 같이 보안기준을 표현하였다. 보안기준은 모델체킹에서 사용되는 속성(property)과 같은 의미로 사용되고 있다.

```
testState1(S,O):-
        validSubject(S),
        isFile(O),
        canReadFile(S,O),
        not(isGroupMember(S,high)),
        O=o1,
        O=o2.

testState2(S,O):-
        validSubject(S),
        isFile(O),
        canWriteFile(S,o3),
        not(isGroupMember(S,low)).
```

그림 4. No Read Up과 No Write Down에 대한 보안기준

보안기준을 $cr$ 이라고 표현했을때, $cr_i$는 보안시스템에서는 발생하지 않아야 하는 속성을 의미한다. SPR도구에서 보안기준은 다음과 같은 형식으로 표현하고 있다.

$$\bigcap_{i \in N} \overline{cr_i} = \text{true}$$

모든 시스템의 상태에서 위와 같이 보안상 문제점을 발생시키는 보안 기준이 발생하지 않았을 경우, 우리는 보안 시스템이 안전하다고 말할 수 있다. 따라서, testState1(S,O)과 testState2(S,O)에 대한 보안기준이 발생하는지 체킹하게 되면, SPR 도구는 [그림 5]와 같은 "spr.rep" 결과 파일을 생성하게 된다.

```
/*
*    SPR report file
* File contains criterion and its results about its safety
*/

testState1(_,_)                    succeeded
testState2(_,_)                    failed
```

그림 5. spr.rep

testState2에 대한 보안기준에 대해 "failed"가 발생했다는 사실은 No Write Down에 대한 보안속성을 위배했다는 것을 의미한다. SPR 도구에서 생성한 프롤로그 기반의 역추적 결과물을 자세히 살펴보면, [표 1]과 [그림 3]에 나타나 있듯이, subject1이 object3에 대해 읽기, 쓰기 권한이 모두 설정되어 있기 때문에 위와 같은 결과가 생성되었음을 알 수 있다. 본 논문의 지면 사정상 접근통제규칙에 대한 SPSL 코드는 생략하였다.

## 5. 결론 및 향후 연구 방향

보안제품을 생산하기 위해서는 보안 개발자들이 보다 손쉽게 보안모델을 추출, 명세하고 해당 요구사항을 검증할 수 있는 정형검증도구의 개발이 절실히 필요한 실정이다. 본 논문에서는 SPR 이라는 정형검증도구를 이용하여 보안 시스템의 안전성을 명세하고 검증할 수 있는 방법을 소개하고 있다. 정형적 설계 및 분석 방법을 통한 고등급 보안시스템의 개발은 결국 시스템의 안전성 및 보안성을 보장하게 되고, 국내 뿐만 아니라 국외 보안시장에서 국가 경쟁력을 키워나가는데 매우 중요한 역할을 차지하게 될 것이다. 물론, SPR 도구의 활용성을 높이기 위해서는 IDS, Firewall, 운영체제 시스템과 같은 보다 실질적이고, 다양한 보안 시스템을 SPSL로 명세하고 검증한 사례에 대한 활용가이드가 필요하다. 향후 연구방향으로는 SPR 도구에 대한 활용가이드 작성과 GUI 기능을 추가하고자 한다.

## 6. 참고문헌


[1] 정보보호시스템 평가·인증 가이드, 2002. 12.

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

[3] M.A.Harrison, W.L.Ruzzo, J.D.Ullman, Protection in Operating Systems. Communications of the ACM. Vol. 19, Num. 8, August 1976.

[4] D.E.Bell, L.J.Lapadula. Secure Computer System: Mathematical Foundations. MITRE Technical Report 2547, Volume II, May 1973.

[5] R.Sandhu, P.Samarati. Access Control: Principles and Practice. IEEE Communications Magazine, 1994.

[6] W. Stallings, *CRYPTOGRAHPY AND NETWORK SECURITY: Principle and Practice*. Prentice-Hall, 1998.

[7] J.Wielemaker, SWI-Prolog 5.2 Reference Manual, http://swi-prolog.org, July 2003.