# SEEN 접근통제 정책관리 GUI 모듈 설계 및 구현

신삼신*, 이재서*, 김정순* 김민수**, 김용민***
*전남대학교 정보보호협동과정,
**목포대학교 정보보호학과,
***전남대학교 전자상거래학과
e-mail:tkatlsdl@lsrc.jnu.ac.kr

# The Design and Implementation of SEEN Graphic User Interface Module for Access Control Policy Management

Sam-Shin Shin*, Jae-Seo Lee*, Jung-Sun Kim*, Minsoo Kim**, Young-Min Kim***
*Interdisciplinary Program of Information Security, Chonnam National Univ., **Division of Information Security, Mokpo National Univ.,
***Division of E-commerce, Chonnam National Univ.

요 약

본 논문에서는 SEEN 보안운영체제 시스템에서 접근통제 정책관리 도구의 설계 및 구현한 내용을 설명한다. 보안운영체제에 대한 지식이 부족한 보안 관리자가 정책을 설정하고 적용하는 데에 많은 어려움이 따른다. 따라서 본 논문에서는 이러한 정책 적용 및 정책 설정의 문제점을 해결하기 위해 접근통제 정책관리 처리에 대한 것을 사용자가 쉽게 파악 할 수 있고 친숙한 그래픽 기반의 형태로 특별한 지식이 없는 사용자와 보안 관리자들에게 시스템을 효율적이고 편리하게 사용할 수 있도록 한다.

## 1. 서론

1)최근 인터넷 환경의 발전에 따라 전 세계에 연결된 개인 컴퓨터와 네트워크에 접근하여 정보를 이용 할 수 있게 되었으며 시스템의 자원과 정보에 대한 정보이용의 편의성을 제공함과 동시에 악의적인 공격에 노출 되어있는 실정이다. 이러한 문제를 해결하기 위해 보안운영체제와 같은 새로운 보안 기술이 필요하며 보안운영체제 시스템에 대한 연구 개발이 꾸준히 진행 되고 있다. 대표적인 보안운영체제로 SELinux(Security Enhanced Linux) 와 SEEN(SEcurity ENtit 보안운영체제를 들 수 있다 [1],[2].

하지만 보안운영체제의 접근통제 정책관리의 복잡성 때문에 정책 설정에 있어서 일반 사용자에게는 친숙하지 않는 것이 사실이다. 사용자는 자신이 관리 할 수 있는 자원에 대한 설정 방법 등을 이해하는 것이 필요한데 이것은 사용자가 거부감이 없이 사용할 수 있게 작성된 접근통제 정책관리 GUI를 통해 해결 할 수 있다.

본 논문의 구성은 2장에서 관련연구에 대해 기술하고 3장에서 SEEN 보안운영체제의 접근통제 정책관리 GUI 모듈의 설계 및 구현을 기술하며 4장에서 시험 및 평가를 하며 5장에서 결론과 향후 연구내용에 대해서 기술한다.

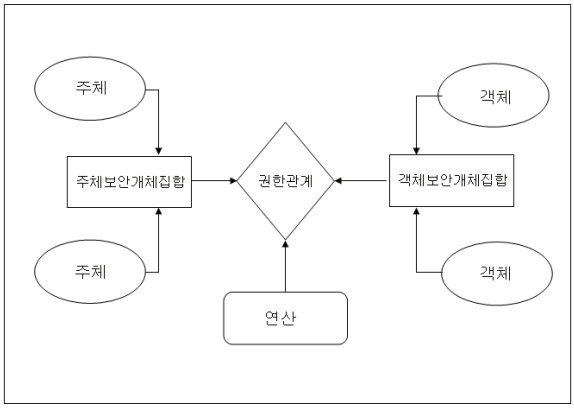
## 2. 관련 연구

### 2.1 SEEN 모델 기본 구조

SEEN(SEcurity ENtity)모델의 기본구조는 보안개체에 해당하는 주체와 객체, 보안개체 집합인 주체보안개체 집합과 객체보안개체 집합, 권한 관계, 연산 등과 권한부여로 구성된다. SEEN 접근통제 모델은 개체, 관계, 속성을 접근통제에 적합하게 보안개체와

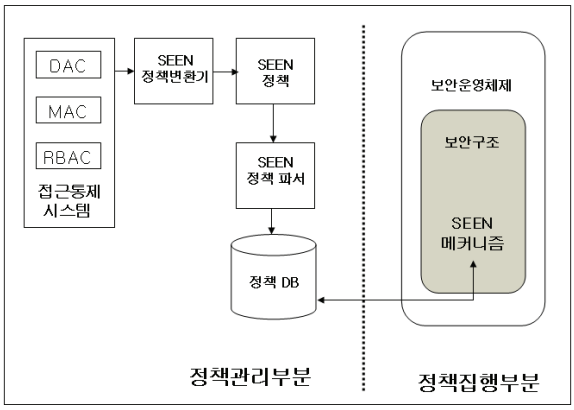
1) 본 연구는 정보통신부 대학 IT 연구센터 육성, 지원사업의 연구결과로 수행되었습니다.

보안개체 집합들의 권한관계로 표현하며 접근 통제는 주체와 객체 그리고 주체가 객체에 행하는 행위(권한 또는 연산)로 나타낼 수 있다. SEEN 모델에서는 실세계의 객체 중에서 접근통제와 관련된 보호대상과 보호주체를 보안개체라 정의한다. 보안개체는 주체와 객체로 구분되며 주체는 객체에 대한 접근을 요청하거나 허가된 행위를 수행하는 행위자에 해당하는 보안개체이다. 현실세계에서는 일반적으로 사용자가 주체가 된다. 객체는 주체가 실제 행위를 수행하는 대상이 되는 보안개체로서 가용한 모든 자원들이 객체가 된다. 경우에 따라서는 주체도 객체의 일종으로 사용되기도 한다. 연산은 주체가 객체에 대해 행하는 일련의 구체적이거나 논리적인 행동으로 객체에 허용된 연산들이다. 다음 그림 1은 SEEN 모델의 기본 구조를 보여준다.




(그림 1 ) SEEN 모델의 기본 구조

## 2.2 SEEN 보안 운영체제 구성도




(그림 2 ) SEEN 보안운영체제 구성도

그림 2는 SEEN(SEcurity ENtity) 모델이 적용된 보안운영체제 구성도를 보여준다. SEEN 모델로 표현된 다양한 접근통제 정책들을 정책변환기를 통하여 SEEN정책으로 변환한다. 변환된 SEEN을 SEEN 정책파서에서 파싱하여 정책DB를 생성한다. 접근통제 보안구조는 정책DB를 참조하여 사용자가 요구한 접근통제 요청에 대해 보안구조내의 접근통제 처리 모듈에서 판단하여 객체에 대한 접근 허가 및 거부가 결정된다[3].

## 2.3 SEEN 정책 기술언어

리눅스 운영체제에서 동작되는 응용서비스 데몬에 대한 SEEN 정책 기술언어로 SEEN 모델의 구성요소인 주체, 객체 및 보안개체 집합 등의 기본 문법을 정의한다. 다음그림 3은 SEEN의 기본 정책기술언어이다. 그림을 통하여 확인 할 수 있듯이 SEEN 정책기술언어를 이용해야 하는데 일반인이나 보안전문가가 정책 기술언어를 이해하여 문법에 맞게 텍스트로 작성해야 하는 어려움이 따른다.

```
// 연산의 정의
○ 연산 정의 문법
operations operation_list_name { operation_list };
○ 사용 예
operations FILE  { open, read, write, execute, close
};

// 보안개체 집합의 선언
○ 보안개체 집합 선언 문법
entity entity_name_list;
○ 사용 예
entity apache_s_entity;
entity apache_o_entity, mysql_o_entity;
// 권한관계의 설정
○ 권한관계 설정 문법
authorize S_S_E O_S_E constraint_list ;
○ 사용예
authorize apache_s_entity apache_o_entity
        FILE(write) ,
        DIR(write, read) ;
//보안개체 집합 부여
○ 보안 개체 부여 문법
grant subject user_name { entity_list };
grant object object_name { entity_list };
○ 사용예
grant subject  webmaster {apache_s_entity};
grant object  /var/www(/.*)? {apache_o_entity};
```

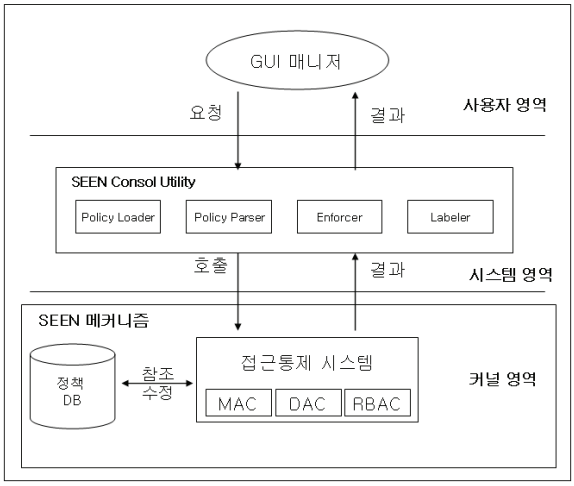
(그림 3 ) SEEN 모델정책기술언어

또한 작성된 기술언어의 문법 오류를 알아내기가 어렵다. 이러한 SEEN 보안운영체제의 특징은 보다 섬세한 접근통제를 가능하게 하지만, 정책기술언어를 통한 접근통제 정책관리에 있어서는 어려움이 있다. 따라서 본 논문에서는 이들의 복잡성을 감소시키고 사용자가 직관적이고 사용하기 쉽도록 접근통

제 정책관리 GUI 도구를 구현 설계하였다[3].

## 3. 사용자 인터페이스 설계 및 구현

### 3.1 GUI 관리도구 전체 구조




(그림 4 ) 정책관리 도구 전체 구성도

위 그림 4는 GUI 접근통제 정책관리 도구의 전체 구성도이다. GUI 접근통제 정책관리 도구는 3 계층 구조를 갖는다. 사용자 계층의 GUI 매니저는 설정된 정책을 각각의 시스템 계층의 모듈에게 전달하여 요청한 작업을 수행한다. 정책관리 GUI 도구의 작업 요청은 정책로드, 정책 파서, 정책 적용, 정책 레이블 모듈을 통하여 커널에 전달되며 각각의 모듈은 해당 작업을 수행한 후 GUI 매니저에게 전달되고 GUI 매니저는 화면으로 결과 내용을 출력하게 된다.

### 3.2 접근통제 정책관리 GUI 요구사항

SEEN 접근통제 정책관리 GUI도구의 요구사항으로는 보안개체 집합(엔티티) 설정기능이다. 이는 보안개체 집합을 관리하는 기능으로 보안개체 집합 추가, 보안개체 집합 삭제, 보안개체 집합 수정 등의 설정기능이 요구되며 주체 설정기능에는 주체보안개체 집합들의 리스트를 출력하고 할당된 사용자 목록과 비할당 목록이 출력 되어야하며, 출력된 사용자 목록을 선택하여 주체의 주체보안개체 집합에 추가/삭제할 수 있도록 구성되어야 한다. 또한 객체 설정 기능은 객체의 보안개체 집합에 정의된 객체보안개체 집합 리스트가 출력이 되며 객체보안개체를 선택하면 객체의 파일, 네트워크 등의 형태로써 리스트로 출력이 되고 출력된 보안개체 집합을 선택하여 객체의 보안개체 집합에 추가/삭제할 수 있도록 구성 되어있다. 마지막으로 주체 보안개체 집합과 객체 보안개체 집합의 권한관계를 설정하고 허용된 연산을 부여하는 기능을 제공되어야 한다. 주체의 보안개체 집합과 객체의 보안개체 집합에 설정된 권한 관계를 삭제하거나 변경할 수 있다.

### 3.3 GUI 요소 설계

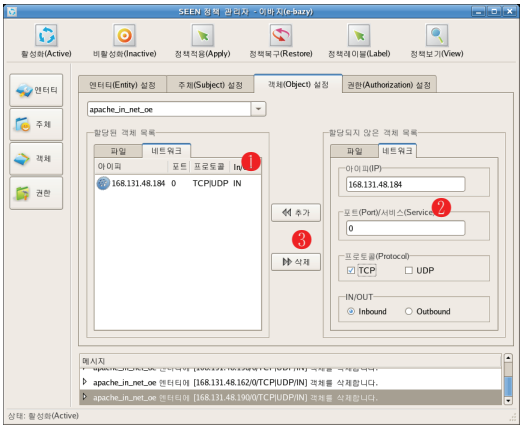
< 표 1 > SEEN GUI 도구 항목별 기능

| 설정 기능 | 설정 세부 기능 |
|---|---|
| 공 통 | - 활성화<br>- 비활성화<br>- 정책 적용<br>- 정책 복구<br>- 정책 보기<br>- 로그 상태 메세지 |
| 보안개체 설정 | - 보안개체 (Entity) 추가<br>- 보안개체 (Entity) 수정<br>- 보안개체 (Entity) 삭제 |
| 주체설정 | - 할당된 주체 선택<br>- 주체 선택<br>- 주체 삭제 |
| 객체 설정 | - 할당된 객체 선택<br>- 객체 선택<br>- 객체 추가<br>- 객체 삭제 |
| 권한 설정 | - 주체 선택<br>- 보안개체 추가<br>- 보안개체 삭제<br>- 보안개체에 할당된 객체 목록<br>- 오퍼레이션 설정 |

표 1는 SEEN GUI 도구의 항목별 세부 기능을 나타낸다. SEEN정책통제 정책관리 GUI 도구의 주요 기능은 SEEN 정책의 기본 구성요소들인 보안개체 집합과 주체, 객체, 연산 및 권한관계를 설정하여 SEEN정책을 작성할 수 있도록 되어 있으며, 작성된 정책을 보안운영체제에 적용할 수 있는 바이너리 형태로 변환하는 기능과 실제 커널에 정책을 로드하는 기능, 설정한 정책을 보기와 잘못된 정책 적용시 복구하는 기능 등으로 구성 되어있다.

### 3.4 구현 및 정책적용 예

Apache는 기본적으로 웹 서비스를 제공하는 서버 데몬으로 다양한 모듈과 보안 설정사항 때문에 세부 기능을 이해하여 정책을 적용하기까지는 매우 어렵다. 그림 5 는 보안정책을 적용하기 어려운 아파치 웹 데몬 서비스에 대해 일반 사용자나 보안 관리자가 직관적이고 사용하기 쉬운 GUI 환경에서 정책을 적용하는 모습을 보여준다. 그림 6은 SEEN 접근통제 정책관리 GUI 도구를 사용하여 아파치 데몬 서비스에 대해 객체들과 연산관계를 참고로 하여 Apache의 기본적인 서비스에 대한 접근통제 정책을 SEEN 정책으로 기술된 예를 나타내고 있다.



( 그림 5 ) Apache 정책적용 화면

```
// 파일명: apache.seen
// 보안 개체 집합 선언
entity apache_s_manager;
entity apache_o_bin, apache_o_conf, apache_o_krb,
       apache_o_passwd, apache_o_home;
...
grant subject  webmaster {apache_s_manager};
grant object  /usr/sbin/httpd {apache_o_bin};
...
// 권한 관계 설정
authorize apache_s_manager apache_o_bin
          constraint FILE(execute);
authorize apache_s_manager apache_o_conf
          constraint FILE(read);
...
```

(그림 6 ) Apache 웹서비스 데몬을 위한 SEEN 정책

## 4. 실험 및 평가

본 논문에서 구현한 SEEN 접근통제 정책관리 GUI 도구를 통하여 정책생성의 정확성 과 편의성 및 효율성에 대해 서비스 데몬 별로 실험을 하였다

< 표 2 > 서비스 데몬 정책생성 시간 비교(분)

| 구분 | apache | smbd | mysqld |
|---|---|---|---|
| 전문가 수동 작성 | 20분 | 15분 | 40분 |
| 일반사용자 도구사용 | 8분 | 4분 | 7분 |

표 2은 보안 관리자가 직접 서비스 데몬별 정책을 수동으로 작성을 하였고 일반사용자는 정책설정의 어려움 때문에 수동 작성을 배제하여 정책관리 GUI 도구를 이용하여 시간을 측정하였다. 다음 표 3은 정책의 정확성을 평가한 것으로 실제로 정책을 설정하거나 작성하여 데몬 서비스를 작동 시켰을 때 서비스정책에 대한 표준정책과 비교를 하여 정책의 정확성을 평가한 것이다.

<표 3 > 서비스 데몬 정책생성 정확성 비교(%)

| 구분 | apache | smbd | mysqld |
|---|---|---|---|
| 전문가 수동 작성 | 70% | 80% | 60% |
| 일반사용자 도구사용 | 90% | 70% | 80% |

표 2, 3을 통하여 접근통제 정책관리 GUI 도구를 이용한 정책 설정 및 관리에 있어서 정확성과 효율성의 신뢰할만한 결과를 보여준다.

## 5. 결론

본 논문에서는 SEEN 보안운영체제 시스템에서 보안정책 설정에 있어서 복잡도 및 접근통제 정책의 관리의 어려움을 감소시키기 위한 접근통제 정책관리 사용자 인터페이스인 SEEN정책 관리 GUI 도구의 설계 및 구현 내용을 설명 하였고 실험과 정책적용의 예제를 통하여 보안정책 적용의 편리함과 효용성을 확인하였다.

보안운영체제 시스템은 아직은 범용화 되어있지 않다. 그의 주요 원인은 사용법과 보안정책의 설정과 적용의 어려움이 주요 원인이기 때문이다. 보안운영체제의 발전과 사용자 위주의 GUI환경의 인터페이스 개발은 보안운영체제 시스템의 개발과 함께 진행되어야 하는 필수 요소가 되어야 할 것이다.

## 참고문헌

[1] Bill McCarty, "SELINUX - NSA's Open Source Security Enhanced Linux", O'REILLY, 2005

[2] Security Enhanced Linux, http://www.nasa.org/selinux

[3] 김정순, "보안 운영체제 정책 유연성을 제공하기 위한 메타 접근통제모델" 전남대학교 2006 박사학위 논문

[4] P. Loscocco, S. Smalley, "Integrating Flexible Support for Security Policies into the Linux Operating".

[5] S. Smalley, Configuring the SELinux Policy, Technical report, NSA, Feb. 2002.

[6] GTK+ The GIMP Toolkit, http://www.gtk.org

[7] Glade - User Interface Designer for GTK+ and GNOME, http://glade.gnome.org