E-BLP Security Model for Secure Linux System and Its Implementation

Jung-Min Kang, Wook Shin, Chun-Gu Park, Dong-Ik Lee

ABSTRACT

To design and develop secure operating systems, the BLP (Bell-LaPadula) model that represents the MLP (Multi-Level Policy) has been widely adopted. However, user’s security level in the most developed systems based on the BLP model is inherited to a process that is actual subject on behalf of the user, regardless whatever the process behavior is. So, there could be information disclosure threat or modification threat by malicious or unreliable processes even though the user is authorized in the system. These problems can be solved by defining the subject as (user, process) ordered pair and by defining the process reliability. Moreover, when the leveled programs which exist as objects in a disk are executed by a process and have different level from the process level, the security level decision problem occurs. This paper presents an extended BLP (E-BLP) model in which process reliability is considered and solves the security level decision problem. And this model is implemented into the Linux kernel 2.4.7.

Keywords: BLP Model, E-Blp Model, Access Control, Reference Monitor, DRC/Dynamic Reliability Check

1. 서 론

정보시스템에 대한 유 · 무형의 대규모 피해를 예방하기에는 공격 시도들이 점차 여러방향의, 분산화, 자동화 및 은닉화의 형태로 나타나고 있으며, 복잡한 리눅스 시스템의 해킹과 해체가 예상되는 것으로 보고되고 있으며 앞으로도 증가할 것으로 예상된다[1]. 이러한 공격들에 대한 정보시스템 보호의 목적은 보호하고 악의적에 대해 인가된 사용자의 접근을 허용하는 데

※ 본 연구는 한국 정보통신부 정보통신 연구개발 혁신허 신기술 개발사업(2004-116-003-2)의 지원을 받았습니다.
† 본 논의는 한국 정보통신부 정보통신 연구개발 혁신허 신기술 개발사업(2004-116-003-2)의 지원을 받았습니다.
‡ 본 연구는 한국 정보통신부 정보통신 연구개발 혁신허 신기술 개발사업(2004-116-003-2)의 지원을 받았습니다.
등급을 프로세스에 그대로 상속하고 있음을 알 수 있다. 이러한 접근방법의 문제점은 프로세스를 전적으로 신뢰할 수 없다는 것에서 기인한다. 즉, 사용자의 보안 등급과 권한이 전달되는 방법을 오류가 내재되어 있으므로, 의도적으로 수행된 악의적인 (malicious) 프로세스에게 그대로 상속할 경우, 시스템 안전성이 저하될 가능성이 있다. 이는 BLP 모델 접근주제를 정의함에 있어서 시스템 사용자와 실제 접근 주제를 정의하는 프로세스를 동일시하도록 단순하게 정의하고 있기 때문이다, 따라서 사용자와 프로세스간 신뢰관계를 모델에 도입함으로써 해결 가능하다. 또한 다중등급 보안 운용체계는 접근 주제가 접근객에서 존재하는 동급화 적 프로세스를 실행할 때 새로운 프로세스를 위한 보안 등급을 부여해야 하며, 접근 주제 접근 간에서 달라질 경우, 보안 등급 결정 문제가 발생하며 정보보안의 목적에 위배되는 결과가 발생한다[12,13]. 이에 다른 논문에서는 프로세스의 신뢰성을 고려하여, 보안 등급 결정 문제를 해결할 수 있는 확장된 BLP(E-BLP) 보안모델을 제안하고 구현한다.

본 논문의 구성은 다음과 같다. 2장은 보안 운용체계가 제공해야 할 서비스 접근주제와 BLP 보안모델에 대해서 기술하고, 3장에서는 접근주제 접근 간의 관계를 정의하며 E-BLP 모델에 대해 설명하고, 4장에서는 접근주제 접근 간의 관계를 정의하며 E-BLP 보안모델에 포함한다. 5장에서는 본 연구의 업계 동향 E-BLP 모델을 이용한 CSRL시스템을 소개하고, 6장에서는 결론 및 향후 연구 계획을 기술한다.

2. 관련 연구

이 장에서는 기존의 운용체계가 제공해야 할 접근주제 서비스와 이를 구현하기 위해 일반적으로 적용되는 BLP 보안 모델에 대해 기술한다.

2.1 접근주제 서비스[14-22]
 접근주제 서비스는 참조 모니터 (Reference Monitor)의 개념을 구현하여 접근 주제 (Subject)와 접근 객 (Object)에 대한 요구를 보안 정책에 의거해 처리가 정해지는 서비스이다. 접근 주제는 접근 객의 사용을 요구하는 작동적인 (active) 시스템 객체 (entity)이며 주로 사용자 또는 프로세스가 이에 해당한다. 접근 객체는 정보를 저장하기 위한 수동적인 (passive) 시스템 객체이며 주로 파일, 프로그램, 디렉토리, 디바이스 등이 해당한다. 이들 주제와 객체는 대부분의 경우 동적 (Dynamic)으로 구분되고, 이들 구분의 정적 (Static) 기준은 없다. 예를 들어 프로세스 객체 동신을 할 때 신호 (signal)를 보내는 프로세스는 접근 주제가 되며, 신호를 받는 프로세스는 접근 객체가 된다. 다중등급 시스템에서 이들 주제와 객체들은 정보의 중요도 (Sensitivity)에 따라 등급화 되며, 부여된 보안 등급은 접근 정책을 위한 정보로 사용된다. 접근

2.2 BLP 보안 모델
BLP (Bell and LaPadula) 보안 모델은 정보의 기밀 분류 환경에서 상위 레벨 정보가 하위 레벨 정보로 전달 하는 다중등급 정책 (MLP: Multi-Level Policy)을 표현하는 최초의 수학적 모델이다. 다음은 BLP 모델에서 시스템의 안전을 위해서 만족해야 하는 특성들이다.

- Simple Security Property (ss-property)로 주제의 보안등급이 레벨의 보안등급을 지배 (dominate)하려면 해당 주제는 read할 수 있다.
- Star Property (e-property)로 주제의 보안등급이 레벨의 보안등급을 지배하려면 해당 주제는 write할 수 있다.
- ds-property로 접근주제 접근 간에서 불일치가 되어 안된다. 즉 주체는 필요한 권한을 부여받은 접근주제를 실행할 수 있다.

당치 여기서 주제의 보안등급이 레벨의 보안등급을 지배할 때, 같은 BLP 보안 모델에서 중요한 정보가 보호되는 접근주제가 다음과 같은 특성들로 갖는다.

1) 정보의 무결성 (Integrity)을 유지하고 비밀성 (Confidentiality)을 만족한다. 다시 말해 악의적인 프로세스에 의해 정보가 유출될 때 변경 (write)될 수 있는 가능성을 최소화하고 주제에 의한 정보의 접근통제를 고려한다.
2) 식별 (Identification)과 인증 (Authentication)과정을 거친 사용자와 악의적인 프로세스를 식별함으로써 중요한 정보에 접근 (read)하여 정보의 부패를 막을 수 있다.
3) 접근주제의 관리적 확장이 가능하지 않는다.
4) 모든 정보에 접근한 다음에 covert channel이 존재할 수 있다.
5) Trusted subject를 요구한다.
6) 실세계에 적용하기에 너무 엄격하고, 불편함으로서 추가적인 보안성을 고려할 수 있다.

2.3 접근주제의 관리적 확장

이 장에서는 시스템 내에서의 실질적인 접근 주제 간의 프로
세스의 안전한 행위를 보장하고, 실행 가능한 접근 제한의 설정에 발생하는 보안 동급 결정 문제를 해결하기 위한 방안을 제시한다.

3.1 프로세스 신뢰성
(그림 1)은 BLP 모델기반의 시스템에서 주제의 객채에 대한 접근을 보여준다. 사용자는 ID/Password와 Level/Categorgy 정보를 통해 식별 및 인증과정을 거친다. 인증 성공 시 사용자를 대신하는 프로세스가 수행되며 그 프로세스는 시스템 내에서 실질적인 접근 주체가 된다. 하지만 시스템 내에서 사용자를 대신하는 프로세스의 정상적인 행위를 강조할 수는 없다. 다시 말해 참조 모니터는 프로세스의 동급 정보만을 가지고 접근 제한에 대한 접근통제를 수행함으로 비 신뢰적인 프로세스의 악의적인 접근 행위를 통제할 수 없다. 확장된 BLP(F-BLP)모델의 기본적인 아이디어는 시스템 내에서의 실질적인 접근 주체인 프로세스의 신뢰성을 보장하는 것이다 [26,27].

(그림 1) BLP 기반 시스템에서의 주제의 객채 접근

3.2 보안 동급 결정 문제
(그림 2)처럼 현재 실행되고 있는 2등급의 Process 2가 디스크상에 동급화된 접근 객채인 3등급의 Program 3을 실행(exe, fork 시스템 호출)하고자 가정하자. 물론 프로그램 실행 시 접근 통제(ss-property 검사)에 의해 높은 동급의 주체는 낮은 동급의 객채를 read/execute할 수 있다. 하지만 이 때 디스크상의 접근 객채의 보안 동급(3등급)과 접근 주체의 보안 동급(2등급)이 서로 다르다. 그렇다면 접근 객채인 Program 3이 접근 주체로서 새로운 프로세스로 부터 수행하는 순간 접근 주체의 보안 동급과 접근 객채의 보안 동급을 어려움 동급을 부여하여 하지 않기 위한 보안 동급 결정 문제가 발생한다. 이 때 2등급의 주체(Process 2)는 Object 1에 대해서 write권한, Object 2에 대해서 read/write권한이 있고, Object 3에 대해서 read권한이 있다. 3등급의 접근 객채(Program 3)는 Object 1과 Object 2에 대해서 write권한이 있고, Object 3에 대해서 read/write권한이 있다.

3.2.1 접근 주체의 동급을 할당하는 경우
(그림 3)은 2등급의 접근 주체 Process 2가 3등급의 Program 3을 실행할 때, 실행된 Program 3의 접근 주체의 보안 동급인 2등급을 부여받는 경우이다. 현재 Process 3은 접근 주체에서의 동급 BLP 모델의 규칙을 적용했을 때, Object 1과 Object 2에 대해서 write 권한이 있고, Object 3에 대해서 read/write 권한이 있다. read/write 권한이 있다. 하지만 Program 3은 설정 시 접근 주체인 Process 2의 보안 동급을 상속받아서 동급 방향의 결과를 가져온다. 그러므로 동급이 항상된 Program 3은 Object 2에 대해서 read가 가능하며, Object 3에 대해서는 write를 할 수가 없게 되었다. Object 2에 대한 read 허용은 정보의 비밀성(confidentiality) 문제를 일으킬 수 있으며, Object 3에 대한 write 허용이 허용되지 않는 것은 가용성(availability)을 떠나는 결과이다.

(그림 2) 접근 객채 실행 시 보안 동급 결정

(그림 3) 접근주체의 동급을 할당하는 경우

3.2.2 접근 주체의 동급을 할당하는 경우
(그림 4)는 2등급의 접근 주체 Process 2가 3등급인 Program 3을 실행할 때, 실행된 Program 3이 자신의 현재의 보안 동급으로 실행되는 경우이다. 현재 접근 주체인 Process 2는 접근 주체로서의 접근 BLP 모델의 규칙을 적용했을 때, Object 1에 대해서 write권한, Object 2에 대해서 read/write권한이 있고, Object 3에 대해서는 read 권한이 있다. 하지만 Program 3이 적용된 Program 3을 실행할 때, 현재 접근 주체인 Process 2의 보안 동급은 2등급 그대로이나, 새로운 생성된 Process 3의 보안 동급은 객채의 보안 동급인 3등급을 상속받아서 동일 사용자에 의한 접근의 적용이 있어서 접근 주체인 Process 3의 보안 동급이 일치하지 않음으로써 정보의 무결성(integrity)을 저해할 수 있다.

3.2.3 보안 동급 결정 문제 해결
등급 결정 문제를 피하기 위해 실행 가능한 접근 객채등급을
공통 영역(본 논문에서는 Common 영역)으로 묶고 이 영역의 접근 제한들은 보안 등급화가 되어 있지 않다가 접근 제한들을 실행할 때 접근주체의 보안 등급으로 상속하는 방 법을 택한다. 이는 로그인 시 사용자의 의도로 설정된 보안 등급을 시스템네의 접근 통제 모듈에 의해 계산적으로 사용하게 하기 위해서이다. 그러므로 등급 결정 문제 자체를 풀 수 있고, 등급 결정 문제에해서 생기는 비밀성, 무결성, 가용성의 문제는 발생하지 않는다. 하지만 일부 실행 가능한 접근 객체들이 실행 시 반드시 무결성을 보장할 수 있는 것은 아니다. 다시 말해 악의적이거나 취약하다고 알려진 프로그램들은 보호해야 할 시스템 객체들(메일, 디렉터리 등) 접근 시 원하지 않는 행위를 할 수 있으므로 분류(본 논문에서는 Public영역)여야 한다. 또한 Public 영역의 접근 주체들은 보안등급이 부여되지 않은 일부 사용자들을 위해 존재한다. 이는 시스템의 가용성을 증대시키며, 접근 제한의 대상을 공유 객체들(Shared Object)으로 설정시킴으로써 등급화된 접근 객체들의 기밀성, 무결성을 보장하기 위함이다. (표 1)은 실행 가능한 접근 객체들의 분류 예를 보여준다.

<table>
<thead>
<tr>
<th>표 1</th>
<th>실행 가능한 접근 객체들의 분류 예</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td>Public</td>
</tr>
<tr>
<td>System Programs</td>
<td>Shell Scripts, Utilities, Commands</td>
</tr>
<tr>
<td>User Programs</td>
<td>Applications, Programs (Editor, Office, PGs)</td>
</tr>
</tbody>
</table>

(그림 5) 분류된 프로그램들의 접근 임계

(그림 5)는 위에서 분류된 프로그램들의 실행 시 접근 임계를 보여준다. Common 영역의 프로그램들은 접근통계(Reduction Monitor)에 의해 Layer 4의 등급화된 객체들을 접근할 수 있으며, Public 영역의 프로그램들은 공유 객체들로 접근이 국한된다. 그리고 두 영역의 접근 주체들은 접근통계 모듈인 Layer 3에서 임의의 프로세스의 악의적인 행위 및 이상적 불량행위를 관리할 수 있는 DRC(Dynamic Reliability Check)에 의해 무결성 검사가 이루어진다. 또한 DRC는 보안 관리자에 의해 실행 가능한 접근 객체들을 분류하기 위한 도구로도 사용된다. 4에는 동일한 프로세스의 신뢰성을 검사하는 DRC의 기능을 $\text{fr}(s)$ 함수로서 모델에 표현했다.

4. 정형적 E-ISP 보안 모델

이 장에서는 3장에서 설명된 개념들을 적용하는 확장된 E-ISP 보안모델의 구성 요소들과 특성 함수를 정형적으로 기술한다.

4.1 구성요소

4.1.1 사용자: $U$

* $U = \text{User Levels} P$ (Categories).
* $P(x)$는 임의의 집합 $x$에 대한 역집합(power set)을 나타낸다.
* $\text{User Levels} = \{\text{Top Secret, Secret, Confidential, Classified, Unclassified, Anonymous}\}$
* Categories = (dept A, dept B, dept C, etc)

Anonymous 해제는 시스템내의 등급화된 사용자가 아닌 외부의 사용자들을 위해 존재하며, 공유(Shared) 개체만 접근하도록 제한한다.

4.1.2 프로그램: $P$

* $P = \text{Common U Public}$

프로그램들은 신뢰적인 프로그램 영역인 Common과 공유 개체로 접근한 것을 허용하는 Public영역으로 나뉜다. 또한 공격적인 프로그램들은 신뢰 있는 경로를 보장하기 위해 참조 모니터에 의해 제어된다.

4.1.3 접근 주체: $S$

* $S = U \times P$
* $U$: 등급화된 사용자들의 집합
* $P$: 프로세스들의 집합

사용자와 접근 통제 사용자들을 대상하는 동일한 주체가 되는 프로세스를 주제로서 정의한다. 이는 동일한 프로세스의 행위를 고려하기 위함이다.

4.1.4 접근 제한: $O$

* $O = \text{Object Levels} P$ (Categories).
* $P(x)$는 임의의 집합 $x$에 대한 역집합(power set)을 나타낸다.
* $\text{Object Levels} = \{\text{Top Secret, Secret, Confidential, Classified, Unclassified, Shared}\}$
* Categories = (dept A, dept B, dept C, etc)

시스템의 가용성과 중요한(sensitive) 개체들(Top Secret ~ Unclassified)의 보호를 위해 Anonymous 등급 사용자의 접근은 Shared 개체로 제한된다.
4.1.5 접근 동작: A
- A = {r, w, a, e}
- r: 재산에 포함된 정보를 읽기.
- w: 정보의 내용을 읽고 쓰기.
- a: 접근 정보를 읽을 수 없고, 새로운 정보 추가.
- e: 실행 가능 접근 과제를 실행.

4.2 시스템 실행: V
- V = (B, M, F): 시스템의 상태를 기술한다.
- B = (S, O, A): 접근 주체 S가 접근 객체 O를 A의 접근 동작으로 접근하는 것을 의미한다.
- M: 접근 주체가 접근 과제의 동작을 덮어쓰는 접근체계로서의 접근 컴퓨터, 열린 접근 과제를 나타낸다. M의 각 요소는 M[s, o]로 표시한다.
- F = (fu, fc, fo, fr): 주체/OBJECT의 동작 정보 함수, 프로그램 명령 검사 함수 및 프로세스 신호 함수의 합(영문 설명) 검사 함수(DRC)이다.
- fu: S → User Levels는 사용자의 최고 보안등급을 검사하는 함수이다.
- fc: S → User Levels는 사용자의 현재 로그인 된 등급을 검사하는 함수이다.
- fo: O → Object Levels는 접근 과제의 보안등급을 검사하는 함수이다.
- fr: S → Boolean는 실행중인 프로세스의 영역을 검사하는 함수이다.

4.3 보안 규칙
- 시스템이 안전하기 위해서만족해야 할 특정된 특성이 있다.

4.3.1 e-sa-property(Extended Simple Security Property)
- if M[s, o] = r, (fp(s) = common) ∧ (fr(s) = true) ∧ (fo(s) ≥ fo(o)).

4.3.2 e*-property(Extended Star-Property)
- if M[s, o] = a, (fp(s) = common) ∧ (fr(s) = true) ∧ (fo(s) ≤ fo(o)).
- if M[s, o] = w, (fp(s) = common) ∧ (fr(s) = true) ∧ (fo(s) = fo(o)).
- if M[s, o] = r, (fp(s) = common) ∧ (fr(s) = true) ∧ (fo(s) ≥ fo(o)).

4.3.3 e-ds-property(Extended ds-Property)
- if (s, o, a) ∈ B, (fp(s) = public) ∧ (fr(s) = true) ∧ (fo(o) = shared) ∧ (a ∈ M[s, o]).

5. E-BLP 보안 모델 기반 CRLS
- 이 장에서는 본 연구원이 개발중인 E-BLP 보안 모델을 적용한 CRLS(CSRL is Secure Linux) 시스템에 대해 기술한다. 현재 개발중인 CRLS 시스템에서는 부서(category) 정보를 제공한다. (그림 6)은 전체적인 CRLS 접근 통제 시스템을 보여준다. 각 장의 설명과는 Appendix로 참고하기 바란다.

5.1 주체/과제 동급화
- 정적 동급인 사용자 주체의 동급과 접근 과제의 파일의 동급은 6(Top Secret)에서 1(Anonymous, Shared)까지 정의한다. 각 사용자의 동급은 SCHD(Security Kernel Database)에 보안관리자에게 의해 정의된다. 접근 과제의 파일의 동급과 실행 시 접근 주체의 프로그램들의 영역정보(공용 또는 공공)를 위해 디스크상의 inode 구조체(include/linux/ext2_fs.h)의 ext2_inode구조체의 여타한 필드(reserved) 영역을 사용했으며, common영역의 프로그램들을 위해서는 정수 10을 public영역의 프로그램들을 위해서는 20을 합당했다. 그리고 과제 동급 정보를 read/write하기 위한 시스템 호출을 추가하고, 보안 관리자를 위해 이를 시스템 호출을 사용하는 프로그램들을 작성했다. 접근 과제의 파일의 동급은 프로세스에게 의해 참조될 때 예외(include/linux/fs.h) 구조체에 추가된 필드(o_level)로 입혀지고, 프로그램의 영역정보는 task_struct(include/linux/sched.h) 구조체에 추가된 필드(domain)로 입혀진다. 결국 E-BLP 모델에서 정의하는 주체 S는 로그인 시 할당되는 사용자의 동급(a_level)과 프로세스 실행 시(exec 시스템 호출) 할당되는 영역정보로서 구성된다(Appendix A, C).

(그림 6) CRLS 접근 통제 시스템

5.2 인증도모
- 식별(Identification)과 인증(Authentication)은 시스템 자원을 보호하기 위한 외부의 1차적인 보호 계층이다.
CSRL 시스템의 인증절차는 다음과 같다. (그림 6)에서 (1) 동급권한 사용자는 자신의 보안 등급(또는 자신보다 낮은 보안등급)으로 시스템에 로그인을 시도한다. (2) 로그인 프로세스는 사용자가 제출한 등급과 그 사용자를 대표하는 uid (User Identifier)를 커널내의 인증모듈에 제시하고, (3) 인증모듈(Auth. Module)은 SKDB(Security Kernel Data Base)에서 보안 관리자에 의해 미리 정의된 uid와 해당하는 사용자의 최고 보안 등급을 검색한 후, 로그인 시 제출된 보안등급이 최고 보안등급보다 낮거나 같으면 인증하기를 로그인 프로세스에 알린다. (5) 시스템에서 인가된 사용자를 대신하고, 로그인 시 사용자의 보안등급을 상속한 Shell 프로세스를 실행한다. 이때 사용자의 보안등급은 라이브러리에서 프로세스 디스크립트인 task_struct 구조체의 새로운 정의된 필드(s_level)에 입력하고, 실행되는 Shell 프로세스의 영역정보로 새로운 정의된 필드(domain)에 입력한다. (6) 보안 등급 문제를 피하기 위해 새로운 생성(fork/进程호출)되는 프로세스들은 사용자의 보안등급을 계속 상속한다.[Appendix B].

5.3 접근 통제 모듈
(그림 7)은 (그림 6)의 접근 통제 모듈의 접근 통제 수행을 위한 도식적인 그림이다. 접근은 복합요소에 대한 요구를 하면 AEF(ACCESS ENFORCEMENT FACILITY)는 그 요구를 받아서 ADF(ACCESS DECISION FACILITY)에 접근검증을 요청한다. ADF는 시스템의 안전을 결정하는 정책(예: DAC, MAC, RBAC)에 의거해 접근 허용/불용을 결정하고 그 결과를 AEF에게 알린다. AEF는 최종적으로 접근 결정을 결정한다(28). 다시 말해 AEF는 접근 통제 메커니즘을 구현하며, ADF는 접근 통제 정책을 구현한다. 이런 정책과 메커니즘 별의 이중은 정책 변경 시 AEF(메커니즘)에 대한 허락정의 변경으로 ADF 정책을 수정할 수 있다는 것이다. 그러므로 다양한 정책을 사용할 수 있는 시스템의 설계 및 구현이 가능하다(29). 

(그림 8) Access Decision Facility
E-BLP 모델 기반CSRL 시스템의 접근 통제는 (그림 6)의 (7)과 같이 주체인 프로세스의 접근 제한에 대한 접근을 요구할 때 수행된다. AEF는 우선 접근을 요구한 주체의 영역(공용 또는 공통)을 판단하고, 접근 동작이 로드 또는 writin을 판단한 후, 각 영역에 해당하는 보안 정책어려물에 계산된 ADF가 MAC 모듈에 접근 제한을 요청한다. 접근 결정을 요청한 보안 ADF는 접근 제한을 판단한 영역별로 e-es-PROPERTY 점자 또는 e-s-PROPERTY 점자를 수행한다. (그림 8)은 모듈으로서 제공되는 ADF MAC 모듈의 접근 결정 방식을 보여준다. 현재는 프로세스의 인증성 점자 투진(DRC)을 설정한다.[Appendix D]

6. 결론 및 향후연구
다중등급 보안 운영체계에서의 기존 접근 제한 방법은 사용자의 인증을 실질적인 주체인 프로세스에 상속함으로써 프로세스 자체의 행위 및 신뢰성을 고려하지 않았다. 또한 실행 가능한 접근 제한의 설정 시 보안 등급 결정 문제가 정보보호의 목적에 부합될 수 있다. 위의 문제점을 해결하기 위해 본 논문에서는 사용자와 프로세스간 신뢰관계를 고려하고, 보안 등급 결정 문제를 해결할 E-BLP보안 모델을 제안하고 결정성 CSR의 시스템에 대해 소개하였다.

향후 연구개발로는 모델에서 제공한 프로세스 신뢰성 점자를 위한 독립적 및 기반적인 연구가, 다양한 정책을 사용할 수 있는 ADF의 설계 및 구현, 감사 기능 등이 있다.

Appendix

3장에서 설명된 E-BLP 모델 기반CSRL 시스템의 실행 결과를 보여준다.
A. 접근 객체 통집합 및 프로그램 영역지정

```
root(0x37) /etc/passwd

The level of the /etc/passwd has been changed to TDP_SEDE
root(0x37) /etc/krb5.conf

The level of the /etc/krb5.conf is TDP SEDE
```

B. CSFL 인증 절차

```
[밥상에서 맥주를 마시면서]

[2x2] Concurrent System Research Laboratory Security Research Group Test-bed

Kernel: 2.6.31 on an 1166
CSFL_LOGIN: jdk
Password: jdk
CSFL LEVEL: 3
Login Success!
Your Current Level is CLASSIFIED
```

C. 실행 가능한 프로그램 실행 시 주체 정보 출력 예

다음은 프로그램 /usr/bin/passwd의 영역 정보를 COM-
MON으로 설정하고, 현재 3등급으로 로그인한 사용자가 /usr/
bin/passwd 프로그램을 실행 시 /var/log/messages의 내용
을 보여줍니다.

```
Sep 21 02:44:50 thesis kernel: The information of the Subject Sm(P) is ...
Sep 21 02:44:50 thesis kernel: The User's Current Level is CLASSIFIED
```

D. 접근 통제

D-1 AEF(Access Enforcement Facility) Snapshot

```
asm linkage ssize_t sys_read(unsigned int fd, char *buf, size_t count)
{
    int Themsis_ADF_result = 0;
    ret = EBADF;
    file = fpget(fd);
    if (file)
        if (file->f_mode & FMODE_READ)
            AEF;
        if (file->f_mode == NULL)
            inode = file->f_dentry->d_inode;
            if ((current->s_level == 0) && (inode->o_level != 0))
                /* Call ADF Module */
                Themis_ADF_result = ADF_read_file(current->domain,
                        current->s_level, inode->o_level);
                if (Themis_ADF_result == 0)
                    printf("You are not allowed to access!");
                return 0;
            }
        }
        
```

D-2 ADF(Access Decision Facility) Module

```
/* This Module contains the ADF(Access Decision Facility) of
CSFL System */
#include <linux/kernel.h>
#include <linux/module.h>
#include <sys/syscall.h>
#include <config/modversions>
#define MODVERSIONS
#include <linux/modversions.h>
#endif

/* Decision functions declaration
Using below names, ADFs call decision functions of this ADF */
extern int (*ADF_read_file)(int domain, int s_level, int o_level);
extern int (*ADF_write_file)(int domain, int s_level, int o_level);
int chlp_read_file(int domain, int s_level, int o_level) {
    printk("chlp_read_file() is called in ADF module(s)!");
    return 1;
}
int chlp_write_file(int domain, int s_level, int o_level) {
    printk("chlp_write_file() is called in ADF module(s)!");
    return 0;
}
```

D-3 테스트 결과

다음은 Classified등급으로 로그인한 사용자가 Top Secret
파일인 /etc/passwd 파일을 cat명령을 이용해서 읽기 권
한을 시도한 경우의 테스트 메시지입니다.

```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```