2001 년 한국정보처리학회 추계 학술발표논문집 제 8 권 제 2 호

다중등급 보안 운영체제에서의 보안 등급 결정 문제

강정민*, 신 욱*, 박춘구*, 이동익*, 이경호**
*광주과학기술원 정보통신공학과
**한국전자통신연구원 전자상거래연구부 SCM 연구팀
e-mail : *{jmkang, sunihill, cgpark, dilee}@kjist.ac.kr
**khleesun@etri.re.kr

Security Level Decision Problem in MLP-based Secure OS

Jung-Min Kang*, Wook Shin*, Chun-Gu Park*, Dong-Ik Lee*, Kyeong-Ho Lee**
* Dept. of Information and Communications,
Kwang-Ju Institute of Science and Technology (K-JIST).
**SCM Team, Department of EC, ETRI.

요 약

대부분의 안전한 운영체제는 주체와 객체에 보안 등급을 부여하여 운영하는 다중등급 정책 (MLP: Multi-Level Policy)을 수용하고 있으며, BLP 모델은 이 정책을 표현하는 검증된 대표적인 모델이다. 하지만 이러한 다중 등급 보안 운영체제들은 접근 주체인 프로세스가 접근 객체로서 존재하는 등급화 된 프로그램을 실행 시 새로운 프로세스를 위한 보안 등급을 부여해야 하는데, 접근 주체와 접근 객체의 보안 등급이 다를 경우 보안 등급 결정 문제가 발생하며 정보보호의 목적에 위배되는 결과가 발생한다. 이에 본 논문에서는 위에 언급된 문제를 해결할 수 있는 방안을 BLP 모델 측면에서 고찰한다.

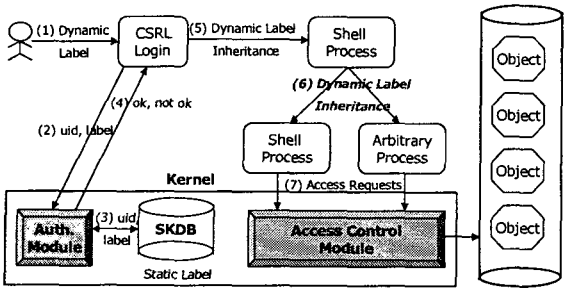
1. 서론

정보보호의 목적은 보호해야 할 객체에 대해 인가된 사용자의 접근만을 허용하는 비밀성(Confidentiality), 인가된 사용자나 인가된 방식에 의해 객체의 수정을 허용하는 무결성(Integrity), 그리고 인가된 사용자에게 객체의 접근을 허용하는 가용성(Availability)을 보장하는 것이다[1]. 최근 정보보호 기술로서 대두된 방화벽이나 침입 탐지 시스템 같은 어플리케이션 수준에서의 정보보호의 노력은 안전한 운영체제의 기반이 없이는 실현될 수 없다[2].

대부분의 안전한 운영체제는 주체와 객체에 보안 등급을 부여하여 운영하는 다중등급 정책(MLP: Multi-Level Policy)을 수용하고 있으며, BLP(Bell and LaPadula) 모델[3,4,5,6,7]은 이 정책을 표현하는 검증된 대표적인 모델이다. 하지만 이러한 다중등급 보안 운영체제들은 접근 주체인 프로세스가 접근 객체로서 존재하는 등급화 된 프로그램을 실행 시 새로운 프로세스를 위한 보안 등급을 부여해야 하는데, 접근 주체와 접근 객체의 보안 등급이 다를 경우 보안 등급 결정 문제가 발생하며 정보보호의 목적에 위배되는 결과가 발생한다[8,9]. 이에 본 논문에서는 위에 언급된 문제를 해결할 수 있는 방안을 BLP 모델 측면에서 고찰한다. 논문의 수월한 진행을 위해 주체/객체에 부여되는 등급/부서 중 부서 정보는 고려하지 않았다.

2. 접근통제 시스템

접근통제 시스템이란 참조 모니터[10,11]의 개념을 구현하여 접근 주체(Subject)의 접근 객체(Object)에 대한 요구를 보안 정책에 의거해 허가/거절하는 시스템이다[12,13,14,15,16,17]. 접근 주체는 접근 객체의 사용

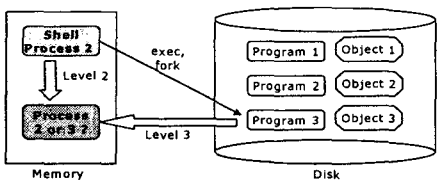


[그림 1] 접근통제 시스템

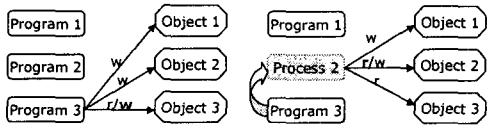
을 요구하는 능동적(active)인 시스템 개체(entity)이며 주로 사용자 또는 프로세스가 이에 해당된다. 접근 객체는 정보를 저장하기 위한 수동적(passive)인 시스템 개체이며 주로 파일, 프로그램, 디렉토리, 디바이스 등이 해당된다. 이들 주체와 객체를 구분하는 뚜렷한 기준은 없으며, 예를 들어 프로세스 간의 통신을 할 때 신호(signal)를 보내는 프로세스는 접근 주체가 되며, 신호를 받는 프로세스는 접근 객체가 된다. 다중 등급 시스템에서 이들 주체와 객체들은 정보의 중요도(Sensitivity)에 따라 등급화가 된다. 주체와 객체에 부여된 보안 등급은 접근 결정을 위한 정보로서 사용이 된다. 접근 통제를 위한 속성을 BLP 모델에서는 다음과 같이 정의하고 있다.

- Simple Security Property: 주체의 보안 등급이 객체의 보안등급을 지배(dominate)하면 해당 객체를 read 할 수 있다.
- *-Property: 객체의 보안등급이 주체의 보안등급을 지배하면 해당 객체에 write 할 수 있다.
- ds-property: 현재의 모든 접근은 접근행렬에 표시되어야만 한다. 즉 주체는 필요한 권한을 부여받은 접근만을 실행할 수 있다.

[그림 1]은 전체적인 접근통제 시스템을 보여주고 있다. (1) 등급화 된 사용자는 자신의 보안 등급(또는 자신 보다 낮은 보안등급)으로 시스템에 로그인을 시도한다. (2) 로그인 프로세스는 사용자가 제출한 등급과 그 사용자를 대표하는 uid(User Identifier)를 커널내의 인증모듈에 제시하고, (3,4) 인증모듈(Auth. Module)은 SKDB(Security Kernel Data Base)에서 보안 관리자에 의해 미리 정의된 uid 에 해당하는 사용자의 최고 보안 등급을 검색한 후, 로그인 시 제출된 보안 등급이 최고 보안등급보다 낮거나 같으면 인증허가를 로그인 프로세스에 알린다. (5) 시스템에서 인가된 사용자를 대신하고, 로그인 시 사용자의 보안등급을 상속



[그림 2] 접근 객체 실행 시 보안 등급 결정



[그림 3] 접근주체의 등급을 할당하는 경우

한 Shell Process 를 실행한다. (6) 접근 주체인 Shell Process 가 시스템 내의 보안 등급화 된 접근 객체인 프로그램을 수행 시 주체의 보안 등급을 상속해준다. 이 때 접근 주체와 접근 객체간의 보안 등급이 다를 경우 실행되는 프로그램을 위해서 어떤 등급을 부여할지 문제가 된다. 다음 장에서는 이 문제에 대해서 자세히 살펴보겠다. (7) 모든 객체에 대한 접근 요구는 위에서 언급한 세가지 보안 규칙을 적용하는 접근통제 모듈(Access Control module)에 의해 통제된다.

3. 보안 등급 결정 문제

이 장에서는 접근 주체인 프로세스가 접근 객체로서 디스크상에 등급화 된 프로그램을 실행 시 (exec, fork 시스템 호출) 발생할 수 있는 보안 등급 결정 문제점을 기술한다.

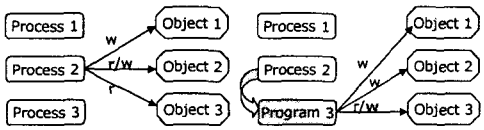
[그림 2] 처럼 현재 실행되고 있는 2 등급의 Process 2 가 디스크상의 접근 객체인 3 등급의 Program 3 을 실행한다고 가정하자. 물론 프로그램 실행 시 접근 통제(ss-property 검사)에 의해 높은 등급의 주체는 낮은 등급의 객체를 read/execute 할 수 있다. 하지만 이때 디스크상의 접근 객체의 보안 등급(3 등급)과 접근 주체의 보안 등급(2 등급)이 서로 다르다. 그렇다면 접근 객체인 Program 3 이 접근 주체로서 메모리로 로드되는 순간 접근 주체의 보안 등급과 접근 객체의 보안 등급 중 어떤 등급을 부여해야 하는지에 대한 보안 등급 결정 문제가 발생한다.

3.1 접근 주체의 등급을 할당하는 경우

[그림 3]은 2 등급의 접근 주체 Process 2 가 3 등급의 Program 3 을 실행할 때, 실행된 Program 3 이 접근 주체의 보안 등급인 2 등급을 부여 받는 경우이다. 원래 Program 3 은 그림 왼쪽에서와 같이 BLP 모델의 규칙을 적용했을 때, Object 1 과 Object 2 에 대해서 write 권한이 있고, Object 3 에 대해서는 read/write 권한이 있다. 하지만 Program 3 은 실행 시 접근 주체인 Process 2 의 보안 등급을 상속 받음으로써 등급 향상의 결과를 가져온다. 그러므로 등급이 향상된 Program 3 은 Object 2 에 대해서 read 가 가능해 졌으며, Object 3 에 대해서는 write 를 할 수가 없게 되었다. Object 2 에 대한 read 허용은 정보의 비밀성(confidentiality) 문제를 일으킬 수 있으며, Object 3 에 대한 write 기능의 불가는 가용성(usability)을 낮추는 결과이다.

3.2 접근 객체의 등급을 할당하는 경우

[그림 4]는 2 등급의 접근 주체 Process 2 가 3 등급인 Program 3 을 실행할 때, 실행된 Program 3 이



[그림 4] 접근객체의 등급을 할당하는 경우

자신의 보안 등급으로 실행되는 경우이다. 원래 접근 주체인 Process 2 는 그림 왼쪽에서와 같이 BLP 모델의 규칙을 적용했을 때, Object 1 에 대해서 write 권한, Object 2 에 대해서 read/write 권한이 있고, Object 3 에 대해서는 read 권한이 있다. 하지만 Program 3 이 실행 시, 주체인 Process 2 의 보안 등급은 Program 3 의 보안 등급으로의 저 등급화 결과를 가져온다. 그러므로 프로세스 3 은 Object 2 에 대해서 read 가 불가능해졌고, Object 3 에 대해서는 write 가 가능해졌다. Object 2 에 대한 read 불가는 가용성을 낮추며, Object 3 에 대한 write 가능은 정보의 무결성(integrity)을 저해할 수 있다.

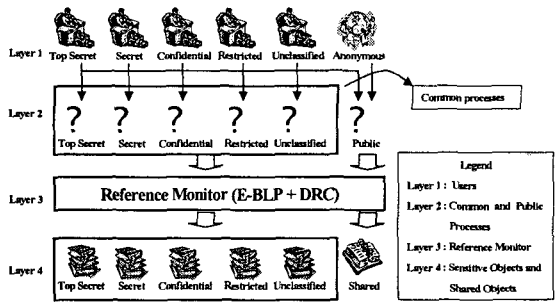
4. 보안 등급 결정 문제 해결

3 장에서 언급한 접근 주체가 접근 객체인 프로그램을 수행 시 등급 결정 문제에서 나타난 비밀성, 무결성, 가용성의 저해는 정보보호의 목적에 위배되는 결과이다. 이장에서는 이를 해결하기 위한 방안을 제시한다.

등급 결정 문제를 피하기 위해 실행가능 한 접근 객체들을 공통의 영역(본 논문에서는 Common 영역)으로 묶고 이 영역의 접근 객체들은 보안 등급화가 되어 있지 않다가 접근 객체들을 실행할 때 접근주체의 보안 등급으로 상속하는 방법을 택한다. 이는 로그인 시 사용자의 의도로 제시된 보안 등급을 시스템내의 접근 통제 모듈에 의해 계속적으로 사용되게 하기 위해서 이다. 그러므로 등급 결정 문제 자체를 피할 수 있고, 등급 결정 문제에서 생기는 비밀성, 무결성, 가용성의 문제는 발생하지 않는다. 하지만 일부 실행 가능한 접근 객체들이 실행 시 반드시 무결성(정상적인 행위)을 보장할 수 있는 것은 아니다. 다시 말해 악의적이거나 취약하다고 알려진 프로그램들은 보호해야 할 시스템 객체들(파일, 디렉토리 등)을 접근 시원하지 않는 행위를 할 수 있으므로 분류(본 논문에서는 Public 영역)되어야 한다. 또한 Public 영역의 접근 주체들은 보안 등급이 부여되지 않은 익명의 사용자들을 위해 존재한다. 이는 시스템의 가용성을 증대시키며, 접근 객체의 대상을 공유 객체들로(Shared Object) 제한 시킴으로 등급화 된 접근 객체들의 기밀성, 무결성을 보장하기 위함이다. [표 1]은 실행 가능

[표 1] 실행 가능한 접근 객체들의 분류 예

	Common	Public
System Programs	Shell Scripts, Utilities, commands	Daemons (httpd, ftpd etc)
User Programs	Applications, Programs (text editor, office programs, etc)	Programs and Scripts developed by users, Hacking programs, etc



[그림 5] 분류된 프로그램들의 접근 진행

한 접근 객체들의 분류 예를 보여준다. [그림 5] 는 위에서 분류된 프로그램들의 접근 흐름을 보여준다. Common 영역의 프로그램들은 접근통제(Reference Monitor)에 의해 Layer 4 의 등급화 된 객체들을 접근할 수 있으며, Public 영역의 프로그램들은 공유 객체들로 접근이 제한된다. 그리고 두 영역의 접근 주체들은 접근통제 모듈인 Layer 3 에서 임의의 프로세스의 악의적인 행위 및 예상치 못한 행위를 판단할 수 있는 DRC(Dynamic Reliability Check)에 의해 무결성 검사가 이루어진다[18]. 또한 DRC 는 보안 관리자에 의해 실행 가능한 접근 객체들을 분류하기 위한 도구로도 사용이 된다.

5. E-BLP(Extended BLP) 보안 모델[18]

이 장에서는 4 장에서 설명된 개념들을 적용하는 확장된 BLP 보안모델의 구성 요소들과 특성 함수들을 정형적으로 기술한다.

5.1 구성요소

5.1.1 사용자 등급: U

- U = { Top Secret, Secret, Confidential, Classified, Unclassified, ***Anonymous*** }

5.1.2 프로그램: P

- ***P = { common, public }***

5.1.3 접근 주체: S

- ***S = (U, P)***

5.1.4 접근 객체 등급: O

- O={ Top Secret, Secret, Confidential, Classified, Unclassified, ***Shared*** }

5.1.5 접근 동작들: A

- A={r, w, a, e}
  - r: 객체에 포함된 정보를 읽기.
  - w: 정보의 내용을 읽고 쓰기.
  - a: 객체 정보를 읽을 수 없고 새로운 정보 추가.
  - e: 실행 가능 접근 객체를 실행.

5.2 시스템 상태

- V = (B, M, F) : 시스템의 상태를 기술한다.
- B = (S, O, A): 접근 주체 ***S = (U, P)*** 가 접근 객체

O 를 A 의 접근 동작으로 접근하는 것을 의미한다.

- M: 접근 주체가 접근 객체에 대해 허가된 동작을 명시하는 접근행렬로서 행은 접근 주체를, 열은 접근 객체를 나타낸다.
- F= (fu, fc, fo, ***fp, fr***): 주체/객체의 등급함수, 프로그램 영역 검사 함수 및 동적 프로세스 신뢰성(무결성) 검사 함수들(DRC)이다.
  fu(s): 사용자의 최고 보안등급 검사 함수.
  fc(s): 사용자의 현재 로그인 등급 검사 함수.
  $\forall s \in S, fu(s) \geq fc(s)$. ($\geq$: dominance relation)
  fo(o): 주체 O 의 보안등급 검사 함수
  fp(s): 실행가능 프로그램의 영역 검사 함수
  fr(s):실행중인 프로세스의 신뢰성 검사 함수.

5.3 보안 규칙

시스템이 안전하기 위해서는 다음의 세가지 성질들을 만족해야 한다.

5.3.1 Extended Simple Security Property (e-ss-property)
if mode=r($\in$M[s,o]), ***fp(s) = common*** and ***fr(s) = true*** and fc(s) $\geq$ fo(o).

5.3.2 Extended Star-Property (e-*-property)
if mode= a($\in$M[s,o]), ***fp(s) = common*** and ***fr(s) = true*** and fc(s) $\leq$ fo(o).
if mode= w($\in$M[s,o]), ***fp(s) = common*** and ***fr(s) = true*** and fc(s) = fo(o).
if mode= r($\in$M[s,o]), ***fp(s) = common*** and ***fr(s) = true*** and fc(s) $\geq$ fo(o).

5.3.3 Extended ds-Property (e-discretionary security)
if (s, o, a) $\in$ B, ***fp(s) = common*** and ***fr(s) = true*** and a $\in$ M[s,o].
if (s, o, a) $\in$ B, ***fp(s) = public*** and ***fr(s) = true*** and ***fo(o) = shared*** and a $\in$ M[s,o].

6. 결론 및 향후연구

본 논문에서는 주체와 객체에 보안 등급을 부여하여 운영하는 다중등급 보안 시스템에서의 접근 주체인 프로세스가 접근 객체로서 존재하는 등급화 된 프로그램을 수행 시 새로운 프로세스를 위한 보안 등급을 부여할 때 생기는 비밀성, 무결성, 가용성의 문제점을 고찰하고, 이를 해결하기 위해 실행 가능한 접근 객체들을 무결성 측면에서 Common 과 Public 영역으로 분류하고, 실행 시 사용자의 보안 등급을 상속 받게 함으로써 일관된 접근통제를 할 수 있게 하였고 제안된 개념을 BLP 보안 모델(E-BLP)에 적용했다.

현재 이 모델을 적용한 안전한 리눅스 운영체제를 구현 중에 있으며 프로세스 신뢰성 검사와 감사 기록을 위한 연구도 진행되어야 할 과제이다.

감사의 글

본 연구는 한국 정보통신부 정보통신 연구센터 육성, 지원사업의 일환으로 수행되었습니다.

참고문헌

[1] NIST, "An Introduction to Computer Security: The NIST Handbook", June 20, 1994.
[2] Peter A. Loscocco, Stephen D. Smalley, Patrick A. Muckelbauer, Ruth C. Taylor, S. Jeff Turner, John F. Farrell, "The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments", Technical report, United Stated National Security Agency (NSA), 1995.
[3] Len Lapadula, "Secure Computer Systems: Mathematical Foundations", MITRE Technical Report, Vol I, 1996.
[4] Len Lapadula, "Secure Computer Systems: Mathematical Foundations", MITRE Technical Report, Vol II, 1996.
[5] John McLean, "The Specification and Modeling of Computer Security", Computer , Volume: 23 Issue: 1 , Jan. 1990.
[6] Carl E. Landwehr, Constance L. Heitmeyer, and John McLean, "A Security Model for Military Message Systems", ACM Trans., Aug. 1984.
[7] Frank L. Mayer, "An Interpretation of a Refined Bell-La Padula Model For the Tmach Kernel", Aerospace Computer Security Applications Conference, 1988.
[8] 홍기용 외, "안전한 운영체제를 위한 MAC 메커니즘의 설계 및 구현", 한국정보과학회 가을학술발표논문집, Vol. 17, No.2, 1990.
[9] Raymond M. Wong, "A Comparison of Secure UNIX Operating Systems", Computer Security Applications Conference, 1990.
[10] DOD 5200.28-STD, "Trusted Computer System Evaluation Criteria", December 1985.
[11] Daniel F. Stern, and Glenn S. Benson, "Redrawing the Security Perimeter of a Trusted System", Computer Security Foundations Workshop VII, 1994.
[12] Dieter Gollman, "Computer Security", Jone Wiley & Sons, 1999.
[13] Edward G. Amoroso, "Fundamentals Of Computer Security Technology", Prentice Hall, 1994.
[14] Ravi Sandhu and Pierangela Samarati, "Authentication, Access Control, and Audit", ACM Computer Survey. 28, 1, Mar. 1996.
[15] Michael V. Joyce, "Access Control and Applications on Trusted Systems", Computer Security Applications Conference, 1992.
[16] Sandhu, R.S., and Coyne, E.J., "Access Control: Principles and Practices", IEEE Communications, 1995.
[17] David Ferraiolo, and Richard Kuhn, "Role-Based Access Control", Proceedings of 15th National Computer Security Conference, 1992.
[18] 강정민, 신 욱, 박춘구, 이동익, "프로세스 신뢰도에 기반한 확장된 BLP 보안 모델과 아키텍쳐 설계", 한국정보과학회 춘계학술대회 논문집,2001.
[19] Bill Neugent, "Where We Stand in Multilevel Security (MLS): Requirements, Approaches, Issues, and Lessons Learned", Computer Security Applications Conference, 1994.
[20] Carl E. Landwehr, "Formal Models for Computer Security", ACM Computing Surveys, Vol. 13, No. 3. 1981.
[21] http://www.rsbac.de
[22] http://www.nsa.gov/selinux/