보안운영체제의 강제적 접근통제(MAC)를 위한 로그 관리자 설계

박춘구° , 신욱, 강정민, 이형효, 이동익
광주과학기술원 정보통신공학과
{cgpark, sunihill, jmkang}@geguri.kjist.ac.kr
{hlee, dilee}@kjist.ac.kr

The Design of a Log Manager for Mandatory Access Control Mechanism of Secure Operating System

Chun-Goo Park° , Wook Shin, Jung-Min Kang, Hyung-hyo Lee, Dong-Ik Lee
Department of Infomation and Communication, K-JIST

요 약

안전한 컴퓨터 시스템 평가기준인 TCSEC(Trusted Computer System Evaluation Criteria)[1] B1급 이상 시스템의 안전한 운영체제들은 강제접근통제(Mandatory Access Control : MAC)메커니즘을 이용하여 정보의 흐름을 제어하고 있다. 하지만 아무리 정확하게 설계된 접근통제 메커니즘이라고 하더라도 시스템 관리자 또는 보안 관리자가 어떻게 시스템의 접근통제 메커니즘을 관리·운영하느냐에 따라 그 시스템의 안전성과 보안에 대한 신뢰도가 결정된다고 할 수 있다. 지금까지 연구되고 있는 대부분의 MAC을 적용한 안전한 운영체제는 접근통제메커니즘의 적용 및 관리·운영상의 보안문제점을 관리할 적당한 방법을 제시하고 있지 않다[4][5][6][7]. 본 논문은 MAC을 적용한 안전한 운영체제의 안전하고 효율적인 관리·운영을 위한 방법으로 LMACM(Log Manager for Access Control Mechanism)을 제안한다.

1. 서론

최근 시스템 보안에 대한 필요성과 중요성이 크게 대두되면서 안전한 운영체제에 대한 연구가 활발히 이루어지고 있다. 기존의 운영체제가 보안 측면에서 가지고 있는 문제점 중의 하나가 임의접근통제(Discretionary Access Control : DAC)메커니즘에만 의존함으로 인하여 정보의 흐름을 제어하지 못한다는 것이다. 그래서 현재 안전한 운영체제를 설계 및 구현하는데 MAC을 적용하여 기존의 문제점을 해결하고자 하는 연구가 나타나고 있다[4][5][6][7].

그러나 지금까지의 연구사례를 검토해 볼 때, 접근통제 메커니즘의 정확한 설계 및 구현에 대한 연구에 비해, 구현된 안전한 운영체제의 보안 취약점 분석 및 안전한 운영체제의 관리, 운영 등에서 발생할 수 있는 보안상 문제점분석 및 해결에 대한 연구는 부족한 실정이다[4][5][6][7]. 접근통제시스템은 그 메커니즘 및 구현이 정확한 경우에도 보안관리자가 접근통제규칙을 올바르게 설정하지 못했을 경우 오류가 발생할 수 있으며, 이를 위해 접근통제규칙을 올바르게 설정하였는지의 여부를 검사해 주는 도구가 제공되어야 한다.

로그정보는 사후감사를 위해 시스템 내에서 발생하는 이벤트를 기록한 정보이며, 로그정보를 이용할 경우 접근통제 규칙이 바르게 설정되었는지의 여부를 보안관리자가 확인할 수 있다. 그러나 현재 리눅스 시스템에 구현된 DAC기반 로그시스템은 이러한 역할을 수행하기에 기능적으로 부족한 면이 있다.

따라서 본 논문은 MAC 메커니즘을 적용한 안전한 운영체제의 보안상 안전하고 효율적인 관리·운용을 위하여 LMACM(Log Manager for Access Control Mechanism)을 제안한다. LMACM은 기존의 로그 메커니즘에 새로운 기능을 추가하여, 시스템에 적용된 MAC 메커니즘의 보안상 문제점 분석 및 안전한 시스템의 운영을 감사한다. 본 논문의 구성은 다음과 같다. 2장에서는 기존의 로그, 감사 메커니즘의 기능에 관하여 언급하고 새로운 로그, 감사 메커니즘의 필요성과 역할에 대해서 언급한다. 3장에서는 안전한 접근통제 메커니즘의 관리·운영을 위한 LMACM의 세부기능에 대하여 설명하며, 마지막으로 4장에서는 향후 LMACM의 연구방향에 대해 기술한다.

2. 로그 및 감사메커니즘 기능

로깅(logging)이란 운영체제나 애플리케이션의 수행 시 발생하는 모든 이벤트를 기록해 두는 과정을 의미한다. 로깅은 다음과 같이 크게 2가지 측면에서 유용하게 사용될 수 있다.

- 관리적 측면[2][3]
  ① 사용중인 프로그램의 오류발생유무를 검사하고, 만약 오류가 발생했으면 발생 시점과 이유를 알 수 있다.
  ② 프로그램의 수행과정이 원하는 순서대로 이루어지는지 확인할 수 있다.
  ③ 사용자가 시스템의 객체(Object)를 접근하는 패턴을 알 수 있다.
  ④ 사용자의 권한부여 오류에 의해 발생할 수 있는 문제를 탐지 할 수 있다.
- 보안적 측면[3]
  ⑤ 보안메커니즘을 불법적으로 우회하려는 반복적인 시도를 알 수 있다.
  ⑥ 사용자의 불법적인 객체(Object)접근을 탐지할 수 있다.
  ⑦ 특정프로그램 및 시스템의 사용시간과 사용자의 정보를 알 수 있다.

그러나 오늘날 시스템 보안의 중요성이 대두되면서 대부분의 운영체제에서는 로깅 메커니즘을 주로 보안적 측면 위주로 사용하고 있다. 다음은 Linux의 대표적인 로그데이터인데 이들로부터 로깅메커니즘이 주로 보안적 측면에서 사용되고 있다는 사실을 알 수 있다.

- /var/log/lastlog : 로그인 정보를 기록하는 파일.
  - 로그인명, 포트, 최근 로그인 시간 등
- /var/log/wtmp : 사용자의 마지막 로그인에 관한 정보가 기록된 파일
  - 사용자, 사용자가 로그인할 때 터미널 또는 서비스 이름, IP 주소, 날짜와 시간, 세션(session)지속 시간
- xferlog : FTP에 관련된 로그를 기록
- httpd/access_log, http/error_log : httpd 관련 접근, 에러 메시지 로그 파일

로깅 메커니즘의 관리적 측면 중 ①,②는 디버깅 도구 등을 이용하여 처리할 수 있지만 관리적 측면의 ③,④는 로깅 메커니즘을 이용하지 않고는 효과적으로 처리하기가 어렵다. 지금까지 로깅 메커니즘은 관리적 측면의 ③,④에 해당하는 부분을 간과해 왔으며[5][6], 특히 DAC과 함께 MAC을 이용하여 커널 수준에서 자원을 안전하게 통제하기 위해 새롭게 설계된 안전한 보안운영체제에서는 ④에 해당하는 보안관련 기능의 분제분석 부분을 충분히 고려해야 한다. 예를 들어, 시스템관리자 또는 보안관리자가 잘못된 접근통제규칙을 적용하였거나 구현상의 접근통제 메커니즘의 허점으로 접근이 허가되지 않은 객체(Object)로의 접근이 이루어지는 경우, ③,④에 해당하는 로그 및 감사메커니즘에 몇 가지 새로운 기능을 추가하여 이러한 문제를 감지하고 해결할 수 있다. 앞에서 언급한바와 같은 기능을 수행하기 위해 로그메커니즘은 다음과 같은 로그데이터를 추가적으로 기록하여야 한다.

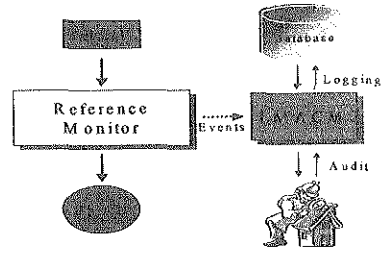
- 호출되는 주요 시스템 콜 이름과 해당 매개변수
- 주체 및 객체의 권한변경사항
- 주체와 객체사이의 신뢰관계의 변경사항

따라서 보안관리자는 기존의 DAC기반의 로그메커니즘에서는 처리할 수 없었던 접근권한의 변경에 의해 발생할 수 있는 보안문제점을 추가적인 로그데이터를 이용하여 운영체제의 안전한 동작의 유·무를 판단할 수 있게 된다.

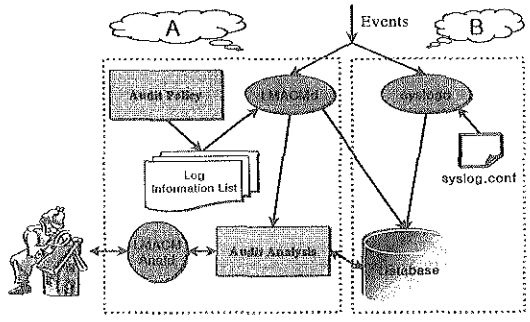
3. LMACM의 세부기능 설계

안전한 운영체제의 참조모니터 (Reference Monitor) 는 시스템 내 모든 주체(Subject)의 객체(Object) 접근시, 접근통제정보(Access Control Information : ACI)를 이용하여 해당 접근행위의 적법성을 검사한다. 참조모니터에서 발생한 이벤트들은 [그림 1]과 같이 기록, 저장된다. 기존 안전한 운영체제연구 사례를 살펴보면 참조 모니터에서 발생하는 모든 이벤트들을 /etc/syslog.conf에 명시된 로깅 정책에 따라 특정디렉토리에 파일 또는 데이터베이스 형식으로 저장하고 있다[2]. 이는 기존의 DAC기반 로그시스템을 그대로 사용하고 있는 것으로[2][8], MAC메커니즘의 적용 및 관리·운영상의 보안문제점을 해결하기 위해서는 [그림 2]의 B부분에 해당하는 기존모듈에 새로운 기능을 수행하는 [그림 2]의 A부분 모듈을 추가할 필요가 있다.

본 논문에서 새롭게 제안한 로그메커니즘 LMACM의 전체적인 기능을 살펴보면 다음과 같다. LMACM은 참조모니터(Reference Monitor)가 생성한 모든 이벤트들을 syslogd 기반 기존 운영체제의 로그메커니즘에 해당하는 [그림 2]의 B부분과 LMACMd(daemon) 기반 안전한 운영



[그림 1] 참조모니터의 기본구조



[그림2] LMACM의 기본 구조

체제의 보안상 안전하고 효율적인 관리·운영을 위한 로그메커니즘에 해당하는 [그림 2]의 A부분을 이용하여 기록하고 관리한다. syslogd 중심의 기존 로그메커니즘은 [9]에서 자세하게 설명하고 있으므로 생략하고, 먼저 LMACMd의 기본 모듈들의 주요기능을 살펴보면 다음과 같다.

- Audit Policy
  다음과 같은 기준으로 감사정책을 설정한다.
  - 보안모델기반
  - 보안 어플리케이션기반
  - 사용자 패턴정보기반
- Logging Information List
  다음과 같은 정보가 감사대상이 될 수 있다.
  - 수행된 프로그램
  - 프로그램의 사용자와 사용 날짜 및 시간
  - 호출되는 주요 시스템 콜과 해당 매개변수
  - 주체 및 객체의 권한변경사항
  - 주체와 객체사이의 신뢰관계의 변경사항
- Audit Analysis
  효과적인 감사를 위하여 다음과 같은 분석자료를 생성한다.
  - 보안모델의 접근통제규칙의 잘못된 적용에 의한 문제점 분석
  - 주체(Subject) 및 객체(Object)의 권한변경에 따른 정보의 흐름 분석 및 오류 점검
  - 사용자의 시스템 사용패턴 분석

LMACMd 중심의 로그메커니즘은 먼저 시스템 관리자나 보안책임자가 시스템의 접근통제메커니즘 변경에 따른 운영체제의 보안기능 안전성 검사를 위하여 먼저 감사정책(Audit Policy)을 결정한다. 안전한 운영체제에 특정 보안모델을 적용했을 경우의 안전성을 검사할 것인지, 특정 보안어플리케이션을 안전한 운영체제에 설치했을 경우의 안전성을 검사할 것인지를 결정한다. LMACMd은 결정된 감사정책(Audit Policy)에 따라 로그정보리스트(Log Information List)에 의해 로그데이터를 선택적으로 결정하고 그 내용들을 데이터베이스에 저장한다. 저장된 자료는 LMACMd의 감사정책(Audit Policy)에 의해 감사분석(Audit Analysis)부분에서 분석·저장되고, LMACM Anald은 분석된 정보를 이용하여 보안기능 안전성에 관한 감사를 수행한다. 예를 들어 특정사용자가 특정객체를 접근할 수 있게 하기 위하여 보안관리자가 그 객체의 접근권한을 변경했을 경우, LMACMd는 해당객체를 접근하는 모든 주체의 로그데이터를 분석함으로써 특정객체의 접근권한 변경이 보안상의 문제점이 있는지 없는지를 감사할 수 있다.

LMACM은 [그림 2]와 같이 기존의 syslogd기반 로그메커니즘에 새롭게 LMACMd기반 로그메커니즘을 첨가함으로써 기존의 로깅메커니즘의 모든 기능은 물론, 안전한 운영체제의 설계, 구현, 운영 등에서 발생할 수 있는 보안상 문제점분석 및 해결을 함께 수행할 수 있는 장점을 가지고 있다. 또한 안전한 운영체제의 보안 메커니즘이 완전하게 구축된 이후에도 사용자의 패턴정보기반 감사정책(Audit Policy)을 이용하여 각 사용자의 시스템 사용패턴을 알아 낼 수 있도록 지원하므로, 유연성을 제공한다.

하지만 LMACM은 기존의 로그메커니즘보다 더 많은 로그데이터를 관리하기 때문에 시스템의 비용 및 성능문제가 발생할 수 있다.

4. 결론 및 향후 연구과제

지금까지 대부분의 안전한 운영체제는 어떻게 접근통제메커니즘을 시스템에 적용할 것인가에 대한 연구가 핵심이었다. 하지만 보안메커니즘의 적용에 따른 보안관련 기능의 문제점분석에 대한 연구도 간과되어서는 안된다. 따라서 본 논문에서 제시한 LMACM이 안전한 운영체제의 커널수준 구현으로 위와 같은 문제를 해결할 수 있다.

향후 연구과제로 LMACM의 커널수준 구현 및 LMACMd과 syslogd과의 연동에 의해 발생하는 시스템의 비용 및 성능문제를 고려한 향상된 로그 관리자를 설계·구현하고자 한다.

5. 감사의 글

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

6. 참고 문헌

[1] National Computer Security Center, "DoD Trusted Computer System Evaluation Criteria", DoD 5200.28-STD, 1985.

[2] Anonymous, "Maximum Linux Security" Sams Publishing, 2000.

[3] National Computer Security Center, "A Guide to Understanding Audit in Trusted Systems, NCSC-TG-001", Version 2, June 1988.

[4] http://medusa.fornax.sk/

[5] http://www.nsa.gov/selinux/

[6] http://www.sctc.com/randt/HTML/dtos.html

[7] Secure Computing Corporation, "DTOS lessons learned report",Part Number 87-0902025A006, June 1997

[8] 최형환, 박태규, 이윤희, "리눅스기반 실시간 감사추적 시스템 구현", 한국통신정보보호학회 종합학술발표회 논문집, Vol.10, No.1, 2000.

[9] Scott Mann, Ellen L. Mitchell, "Linux System Security : The Administrator's Guide to Open Source *Security Tools*, *Prentice Hall PTR*, 2000

[10] M. Bishop, C. Wee, J. Frank , "Goal Oriented Auditing and Logging", IEEE Transactions on Computing Systems, 1996