오픈 소스 보안 운영체제의 성능평가에 대한 연구

홍철호*, 고영웅*, 김영필*, 신용녀**, 유 혁*

*고려대학교, 컴퓨터학과
**한국정보보호진흥원

A Study on Performance Evaluation of Open Source Secure Operating Systems

Cheol-Ho Hong*, Young-Woong Ko*, Young-Pill Kim*,
Yong-Nyu Shin**, Chuck yoo*

*Department of Computer Science and Engineering, Korea University
**Korea Information Security Agency

요 약

근래에 범용 운영체제의 보안성을 강화하기 위해 접근 통제, 침입 탐지 그리고 감사 등과 같은 새로운 보안 기능이 계속해서 추가되고 있다. 추가된 보안 기능에 의해서 어느 정도 시스템 성능에 변화가 생기는지는 보안 운영체제를 도입하려는 사용자들에게 중요한 선택 기준이 되고 있다. 그러나 현재 보안 운영체제들에 대한 성능 분석 자료는 범용 운영체제와 동일한 방법으로 평가된 것들이 대부분이며 여러 보안 운영체제들의 성능을 객관적으로 비교해 놓은 자료 또한 없는 상태이다. 본 논문에서는 잘 알려진 보안 운영체제들을 대상으로 범용 성능 평가 도구를 이용하여 성능 분석 지표를 추출하였다. 본 논문의 결과는 보안 운영체제의 성능 평가를 위해 유용하게 사용될 수 있다.

I. 서론

전통적인 운영체제는 네트워크로 연결된 개방 시스템에 대한 심각한 고려 없이 설계되어 있기 때문에 해커 또는 바이러스의 공격에 노출되어 있었으며 공격을 받고 난 후에 필요한 대책을 세우는 수동적인 방식이었다. 따라서 외부로부터의 위험요소 및 내부적 정보 남용과 같은 문제점에 대해서 적절한 해결책을 제시해 줄 수 있는 운영체제가 필요하게 되었으며 그 결과로 보안 운영체제가 개발되었다.

보안 운영체제란 운영체제 상에 내재된 각종 보안결함을 제거하고 안전하게 시스템을 보호해 줄 수 있는 접근통제, 사용자 인증, 감사 추적 및 침입 탐지 등의 보안기능이 추가된 안전한 운영체제를 의미한다[1].

보안 운영체제는 보안에 대한 문제를 해결해 줄 수 있는 주요한 해결책이다. 그러나 실제적으로는 널리 사용되고 있지 않고 있다. 주된 이유를 몇 가지 언급하자면 다음과 같다. 첫째, 아직까지 보안 운영체제의 중요성에 대한 인식이 널리 확산되지 못하였다는 것을 언급할 수 있다. 두 번째는 보안 운영체제를 사용했을 때 기존의 응용 프로그램과의 호환성이 떨어진다는 것을 언급할 수 있다. 세 번째는 보안 운영체제를 도입했을 때, 시스템 관리를 하기 위한 교육이 필요하며, 새로운 환경에 대해서 부담감을 가지는 관리자 및 사용자들은 기존의 시스템을 그대로 사용하려는 성향을 보이게 된다. 마지막으로 중요한 이유 중의 하나는 보안 운영체제에 대한 성능이 제대로 입증되지 않았다는 사실이다. 따라서 보안 운영체제가 보편적으로 사용될 수 있게 하기 위해서는 보안 운영체제의 성능에 대한 분석이 필수적이다.

하지만 현재까지 나온 대부분의 보안 운영체제 성능 평가 자료는 기존의 마이크로 벤치마크나 매크로 벤치마크 프로그램을 사용하여 측정한 것들이 대부분이다. 기존의 벤치마크 프로그램들은 범용 운영체제의 성능을 측정하기 위해 범용 운영체제에서 중요하게 인식되는 성능에 대해 시나리오를 세우고 구현된 것들이다. 따라서 보안 운영체제에서는 보안 기능을 추가함으로써 생길 수 있는 성능 변화에 대해 새롭게 시나리오를 작성하고 그에 따라 성능 평가를 해야 한다.

본 논문은 리눅스 쪽의 접근통제 프로젝트 중 대표적인 SELinux[2], LOMAC[3] 그리고 RSBAC[4]을 대상으로 성능 평가 연구를 진행하였다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 성능 분석 대상인 SELinux, LOMAC, RSBAC에 대해 기술한다. 3장에서는 보안 운영체제의 성능 분석 절차를 언급하며 4장에서는 범용 운영체제 성능 평가 도구인

이 연구는 한국정보보호진흥원의 국제협력사업에서 지원받았습니다.

lmbench[5]를 이용하여 SELinux, LOMAC, RSBAC의 보안 기능 추가에 의한 오버헤드를 측정하고 보안 성능 지표를 추출하였다. 마지막으로 5장에서 결론을 맺는다.

II. 관련연구

1. SELinux

SELinux는 NSA(National Security Agency)의 주도하에 여러 연구소에 의해 구현된 보안 리눅스 운영체제이다. SELinux에는 접근통제 중 강제적 접근통제인 MAC(Mandatory Access Control)이 구현되어 있는데 전통적인 강제적 접근통제의 구현사항보다 더 진보된 것이다.

SELinux는 Flask[6] 구조를 가져왔으므로 SELinux의 보안 구조는 Flask 구조와 흡사하다[2]. Flask 구조의 특징은 어떠한 운영체제에도 적용 가능하게끔 보안 구조만을 정의하고 있다는 것이다. 그리고 유연한 보안 정책의 지원을 위해서 보안 정책과 보안 정책을 수행하는 보안 메커니즘과의 엄격한 구분을 두어서 최대한의 유연성을 보장해 주고자 한다. 아래 그림 1은 관련 구조를 나타내고 있다.

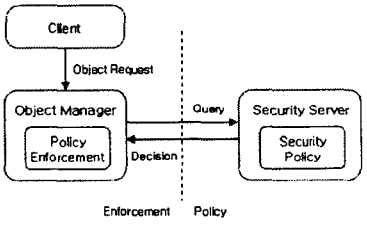

그림 1: SELinux의 구조

그림 1에서 주체인 클라이언트로부터 객체에 대한 접근 요청이 들어오면 객체 관리자가 보안 서버에 객체에 대한 접근 권한 여부를 물어보게 되고 보안 서버는 그 객체에 대한 접근 결정을 객체 관리자에게 넘겨주게 된다. 객체 관리자는 접근 결정에 의해 보안 정책을 수행하게 된다[2].

2. LOMAC

LOMAC은 일반 리눅스 사용자가 불편 없이 사용할 수 있는 MAC 기능을 제공하는 것을 목표로 하고 있다. LOMAC은 로우 워터마크[3] MAC 통합 보호 형식을 제공하며 LKM(Loadable Kernel Module)을 사용하여 구현되었다. LKM은 로더블 모듈로 불리며 리눅스와 같은 시스템에서 사용하는 커널 확장 기능이다.

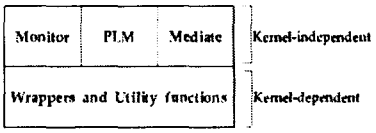

그림 2: LOMAC의 구조

그림 2는 LOMAC LKM의 구조를 보여주고 있다. LOMAC은 커널의 시스템 콜 인터페이스에 자신을 인터포지션 시키는데 커널 백도어처럼 시스템 콜 테이블의 내용을 LOMAC 래퍼(wrapper) 루틴의 주소로 교체한다[7]. 따라서 시스템 콜이 호출되면 LOMAC의 래퍼 루틴이 호출된다. 래퍼 루틴은 시스템 콜 인자인 객체와 시스템 콜을 부른 주체의 접근 관계를 PLM이라는 접근 제이 데이터베이스에서 얻어와 접근 여부를 결정하게 된다. 그 과정은 Mediate 모듈에서 수행되며 Mediate 과정 후에 원래 시스템 콜이 호출된다. 시스템 콜 후출 후에 변경된 접근 제어 내용은 Monitor 모듈에서 수정된다[3].

3. RSBAC

RSBAC은 Amon Ott에 의해서 개발되었으며 몇 가지 접근통제 모델들을 동시에 사용할 수 있게 하는 특징이 있다. 또한 여러 모듈들이 실행되는 중간에도 LKM을 이용하여 동적으로 적재될 수 있으며 모든 접근통제 결정 및 사용자 계정, 시스템 상황 등이 로그에 기록되는 장점이 있다. RSBAC에 구현된 모델은 BLP모델, 기능 통제(Functional Control), 보안 정보 수정(Security Information Modification), 비밀 모델(Privacy Model), 멀웨어 스캐닝(Malware Scanning), 인증(AUTH), 역할 적합성(Role Compatibility) 등이 있다[4].

III.보안 운영체제의 성능 분석 절차

범용 운영체제의 성능은 잘 알려진 마이크로 벤치마크나 매크로 벤치마크 프로그램을 통해 평가되고 있다. 이러한 벤치마크 프로그램은 범용 운영체제에서 성능 상 중요하게 여겨지는 부분을 중심으로 시나리오를 작성한 후 구현된 것이다. 따라서 보안 운영체제에는 보안 운영체제에 적합한 시나리오가 개발되어야 하고 그에 따른 평가 도구로 성능 평가가 진행되어야 한다.

본 논문에서는 그림 3과 같은 방법으로 보안 운영체제의 성능 평가 및 분석을 계획하였다.

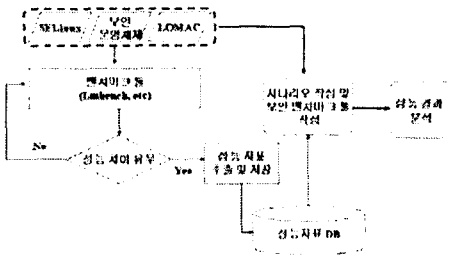

그림 3: 보안 운영체제의 성능 분석 절차도

□한 시스템 내에 범용 운영체제 및 보안 운영체제를 설치한 후 lmbench와 같은 범용 운영체제 벤치마크 툴을 이용해 보안 운영체제의 보안 기능 추가로 인한 오버헤드를 측정한다. lmbench 프로그램은 마이크로 벤치마크 도구로서 시스템의 가장 기본적인 함수 단위로 성능을 측정하여 준다. lmbench는 지연과 대역폭을 측정하는데 중점을 두고 있다[5].

□오버헤드가 생긴 항목에 대해서는 보안 운영체제의 성능 지표로 추출하고 성능 평가 시나리오를 위한 자료로 사용한다. 가령 A, B, C 세 가지 보안 운영체제의 파일 시스템 지연 결과가 범용 운영체제에 비해 모두 낮게 나왔다면 파일 시스템 지연이 보안 운영체제의 성능 지표가 될 수 있다.

□추출한 성능 지표 및 보안 정책의 변화에 기초해 성능 평가 시나리오를 작성한다. 보안 운영체제는 여러 단위 기능의 조합들로 구성되어 있고 단위 기능이 여러 번 수행될수록 보안 운영체제 내에 구현되어 있는 캐쉬 및 보안 정책 수행 과정 등의 영향으로 보안 운영체제마다 성능이 달라질 수 있다. 따라서 보안 운영체제를 위한 평가 시나리오는 보안 기능으로 인한 성능 변화를 종합적으로 평가할 수 있어야 한다.

□성능 평가 시나리오를 보안 벤치마크 도구로 구현한다. 보안 벤치마크 도구를 이용해 성능 평가를 하고 성능 결과를 분석한다.

본 논문에서는 여러 보안 운영체제를 대상으로 성능 지표를 추출하는 부분까지 진행하였다.

IV. 보안 운영체제 오버헤드 분석 및 성능 지표 추출

1. 보안 운영체제 오버헤드 분석

성능 분석에 사용된 범용 리눅스 커널은 2.4.7 버전과 2.2.0 버전이며, SELinux와 RSBAC은 2.4.7 버전에 LOMAC은 2.2.0 버전에 각각 설치하였다. 실험 결과의 정확성을 위해 실험에 사용된 범용 운영체제와 보안 운영체제를 같은 디스크 내에 파티션을 나누어 각각 설치하였다. 성능 분석에 사용된 하드웨어의 시스템 사양은 Pentium III 500MHz, 256M RAM의 데스크탑 PC이다. 우선 지연에 대한 실험을 하였고 다음 대역폭에 대한 실험을 진행하였다.

오버헤드 분석에 사용된 벤치마크 프로그램은 lmbench를 사용하였으며, 운영체제의 각 요소별 성능상의 차이를 보여줄 수 있는 오퍼레이션들을 선정하여 실험하였다. 본 실험에서 선정한 오퍼레이션은 파일 시스템 지연, 통신 대역폭, 통신 지연, 문맥전환 지연, 시스템 호출 지연 등이다. 보안 운영체제의 보안 정책은 기본적으로 제공하는 보안 정책을 이용하였다. 여러 실험 결과 중 아래에 대표적인 몇 가지 실험결과를 보여주고 있다.

1) 파일 시스템 지연

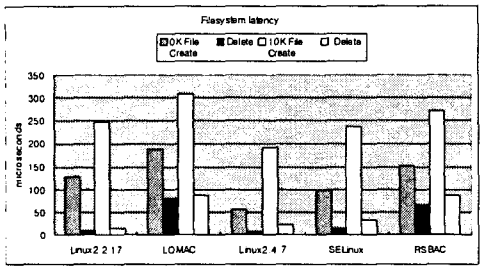

그림 4: 파일 시스템 지연 결과

모든 보안 운영체제의 파일 시스템 지연 시간이 3배까지 높게 측정되었다. 이 실험은 매초 해당되는 파일 사이즈만큼의 파일 개수를 생성하고 지우는데 걸리는 시간을 측정한 것이다. 보안 운영체제가 파일을 생성 및 삭제할 때 파일시스템에 추가적으로 기록하는 보안 레이블과 같은 보안 요소에 의해 추가되는 시간이 어느 정도인지를 보여주고 있다. RSBAC의 파일 시스템 지연이 가장 높게 나왔는데 RSBAC의 활성화된 보안 모델이 다른 보안 운영체제보다 많기 때문이다.

2) 통신 지연

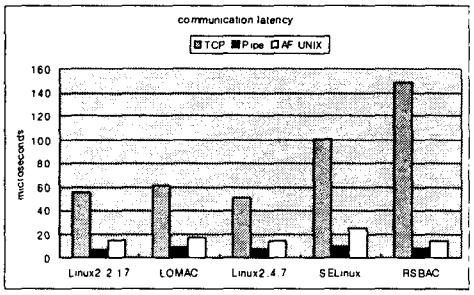

그림 5: 통신 지연 결과

이 실험결과는 보안기능의 추가로 통신 지연시간이 어느 정도 증가되는지 보여주고 있다. 이 실험은 두 프로세스 혹은 클라이언트와 서버 간에 토큰을 서로 주고받는 왕복 시간을 측정한 것이다. LOMAC의 경우 세 가지 통신 지연에 모두 약간의 오버헤드가 있었다. SELinux와 RSBAC의

경우 TCP에서 3배, pipe와 AF_UNIX에서는 2배까지의 증가율을 보여주고 있다.

3) open/close 시스템 호출 지연

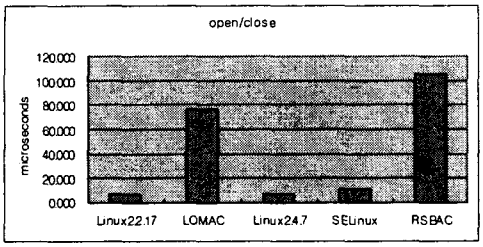

그림 6: open/close 시스템 호출 지연 결과

이 실험은 open/close 시스템 호출에 걸리는 시간을 측정한 것이다. open/close의 과정 중 여러 번의 접근 제어 확인 과정을 거치게 되므로 모든 보안 운영체제의 지연이 높게 나왔다. 특히 LOMAC, RSBAC에서 지연시간의 변동 폭이 크게 나타났다.

2. 성능 평가를 위한 지표 추출

보안 운영체제의 오버헤드를 분석하기 위해 lmbench를 이용해 실험한 결과 개별 보안 운영체제의 구현 방법 및 최적화 기법에 따라 다양한 성능의 분포를 얻을 수 있었다. 전체적으로 파일 및 TCP에 관련된 지연이 높게 측정되었으며 메모리 모듈 및 대역폭에 대해서는 범용 운영체제와 거의 차이가 없었다.

본 논문에서는 앞선 실험 결과로 파일 시스템 지연, TCP 통신 지연, null 시스템 호출 지연, open/close 시스템 호출 지연, stat 시스템 호출 지연, 입출력 지연, 문맥전환 지연, 프로세스 생성 지연 등을 보안 운영체제의 성능 지표로 추출하였다.

V. 결론

본 논문에서는 보안 운영체제의 성능을 측정하기 위한 방법을 제시했으며 그 방법에 따라 보안 운영체제가 가지고 있는 오버헤드를 성능 지표로 추출하였다.

본 논문에서 보이고 있는 분석 결과는 오픈 소스 보안 운영체제인 SELinux, LOMAC, RSBAC에 한정되어 있으나 향후 분석 대상을 다른 오픈 소스 보안 운영체제 및 상용 보안 운영체제로 확대할 예정이다. 또한 객관적인 성능을 분석할 수 있는 시나리오를 개발하고 성능 평가 프로그램을 작성한 후 보안 운영체제의 성능을 분석하는 것을 목표로 삼고 있다.

참고문헌

[1] 한국정보보호진흥원, 안전한 OS 개발 선행 연구, 1998. 12.

[2] Peter Loscocco, "Integrating Flexible Support for Security Policies into the Linux Operating System," 2001 USENIX Annual Technical Conference, 2001.

[3] Timothy Fraser, "LOMAC: MAC You Can Live With," 2001 USENIX Annual Technical Conference, June 2001.

[4] Amon Ott, "Rule Set Based Access Control (RSBAC)," Waardenburg, 14th of September 2001.

[5] Larry McVoy and Carl Staelin, "lmbench: Portable Tools for Performance Analysis," In Proc. of the USENIX 1996 Technical Conference, January 1996.

[6] Ray Spencer, Stephen Smalley, Peter Loscocco, et al, "The Flask Security Architecture: System Support for Diverse Security Policies," In Proceedings of The Eighth USENIX Security Symposium, August 1999.

[7] 홍철호, 고영웅, 김영필, 유혁 "커널 백도어 모듈 탐지 및 차단에 대한 연구," 정보처리학회 학술대회, 2002년 4월.