접근제어 방식이 유닉스시스템 성능에 미치는 영향에 대한 연구

정창성*, 이광현*, 이회선*
*(주)티에스온넷
e-mail:csjung@tsonnet.co.kr

A Study of Influence on System Performance due to Access Control Mechanism in UNIX System

Chang-Sung Jung*, Kwang-Hyun Lee*, Hoi-Sun Lee*
*Research Center, TSonNet Co.,Ltd.

요 약

정보 보호의 목표는 정보자산의 기밀성, 무결성, 가용성을 보장함으로써 정보시스템의 신뢰성과 안전성을 확보하고 이를 통하여 기관이나 조직에서 추구하는 사업에 대한 영속성 보장의 기반을 제공하는 것이다. 접근 제어는 정보 보호의 목표인 기밀성, 무결성을 하기 위한 수단으로 많이 사용된다. 즉, 인가받지 않은 주체에게는 접근을 허용하지 않고, 인가된 주체에 대해서는 신뢰성 있는 정보를 제공하기 위해 정보에 대한 접근 제어를 한다. 그러나 가용성 측면을 무시하고 기밀성과 무결성만을 지나치게 강조할 경우 사용자에게 제공되는 정보는 이미 과거의 정보가 되어 아무런 가치가 없을 수도 있다. 이에 본 논문에서는 정보 보호의 3 대 목표를 모두 만족하는 접근제어 시스템을 구축하는데 있어서 바람직한 방향을 제시하고자 한다.

1. 서론

국가적으로 중요한 정보를 컴퓨터 및 네트워크 시스템의 적절한 보안 대책 없이 외부 네트워크를 통하여 유통 처리되고 있어 보안 문제가 심각하게 대두되고 있다. 이에 해커 등의 불법 사용자로부터 정보를 보호하기 위해 강제적 접근제어(Mandatory Access Control)[2]가 연구 대상이 되어 여러 가지 보안 모델이 제안되고 있다. 종래의 유닉스 운영체제는 사용자 소유의 파일에 대해 허가 비트(Permission Bit)를 설정하여 그룹 내의 사용자 또는 다른 사용자에게 읽기, 쓰기 및 실행 권한을 임의적으로 허가하는 제어를 사용한다. 이러한 접근제어 방법을 임의적 접근제어(DAC : Discretionary Access Control)[1]라고 한다. 오늘날에는 상기와 같은 임의적 접근제어만으로는 정보 보호가 미약하다. 이에 본 논문에서는 기존 보안 모델들 중에서 그 보안성이 비교적 우수하다고 평가되고, 상용화된 보안 시스템에서 널리 사용되고 있는 BLP(Bell & LaPadula)[2, 4, 5] 보안 모델을 유닉스 운영체제의 커널에 적용하여 강제적 접근제어를 구현하였다. 임의적 접근제어는 기존 유닉스 시스템에서 사용한 접근 허가 비트를 그대로 수용하였고, 강제적 접근제어는 수정된 BLP 모델을 적용하여 주체와 객체 사이의 보안 속성 관계를 통한 접근제어를 적용하였다. 구현된 접근제어 시스템이 임의적 접근제어만을 이용하여 접근제어 한 경우, 파일시스템에 보안속성을 직접 부여하여 객체에 대한 접근제어를 수행하는 경우, 파일시스템을 이용하지 않고 접근제어 테이블을 이용하여 접근제어를 수행할 경우에 대하여 각각 시스템의 성능에 미치는 영향을 알아보고자 한다.

본 논문에서는 상용 OS(AIX, HP-UX, Linux, Solaris, Tru64 등) 중에서 일반인들에게 많이 알려져 있고, 인터넷상에 비교적 많은 정보가 공개되어 있는 Solaris 8 x86 Platform 버전을 대상으로 하였다.

2. 시스템 접근 제어

접근제어는 로컬 및 네트워크의 특정자원에 대하여 접근권한이 있는지를 검사한 후 접근여부를 결정함으로써 불법 침입자에 의한 불법적인 자원 접근 및 파괴를 방지할 수 있다. 접근제어 정책으로는 DAC 과 MAC 이 있다. DAC 은 각 주체에 대하여 시스템 객체들에 부여된 권한을 명시하는 권한부여 규칙을 요구한다. 접근 요청은 DAC 메커니즘에 의하여 검사되고 권한부여 규칙이 존재하고 해당접근이 검증되는 주체에게만 허가된다. (그림 1)은 DAC 의 개념적 구조를 나타내고 있다.

(그림 1) DAC 을 통한 접근제어

객체에 포함된 정보의 비밀성(레이블로 표현된 허용등급)과 이러한 비밀성의 접근 정보에 대하여 주체가 갖는 정형화된 권한(즉, 접근허가)에 근거하여 객체에 대한 접근을 제한하는 방법을 MAC 이라고 한다. MAC 은 하위 비밀등급의 객체로 정보의 흐름을 방어하기 때문에 흐름-제어 정책으로 정의될 수 있다. 데이터에 대한 접근은 주체와 객체가 갖는 보안등급의 정의를 통한 강제적인 정책에 의하여 결정된다. (그림 2)는 MAC 의 개념적 구조를 나타낸다.

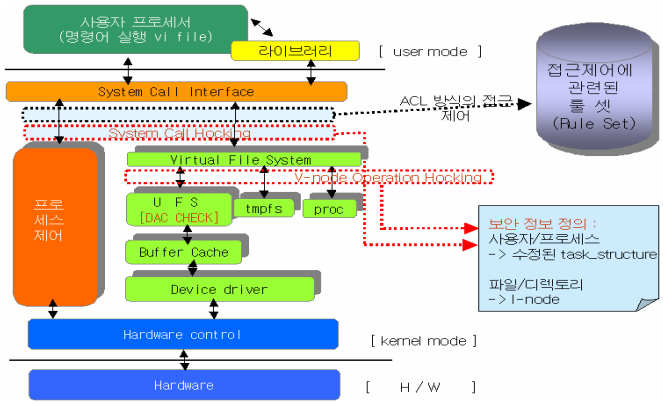
(그림 2) MAC 을 통한 접근제어

Solaris 는 TCSEC[2, 5] C1 등급의 패스워드와 DAC 만을 지원하기 때문에 보안상의 어려움이 있다. 본 논문에서는 TCSEC B1 등급의 특징인 보안레이블을 통한 강제적 접근제어 기능과 접근제어 목록(ACL)을 통한 강제적 접근제어 기능을 추가해서 보안을 유지할 수 있도록 하였다.

3. 강제적 접근제어 시스템 설계

31. 접근제어 구조

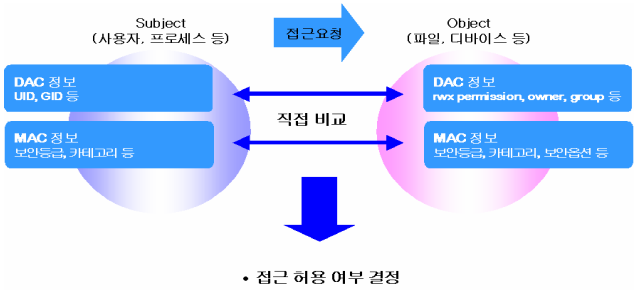
유닉스 시스템에서 사용자가 파일을 읽으면 Open 과 같은 시스템 콜이 호출되면서 파일의 정보가 저장된 I-Node 를 참조한다. 이 때 Open 시스템 콜에 대하여 후킹 기법을 이용하여 인터럽트를 동작시킨 다음 강제적 접근제어에 필요한 조건들을 적용한다.



(그림 3) 유닉스 시스템의 실행 구조

3.2 ACL 테이블 구조 기반의 접근제어

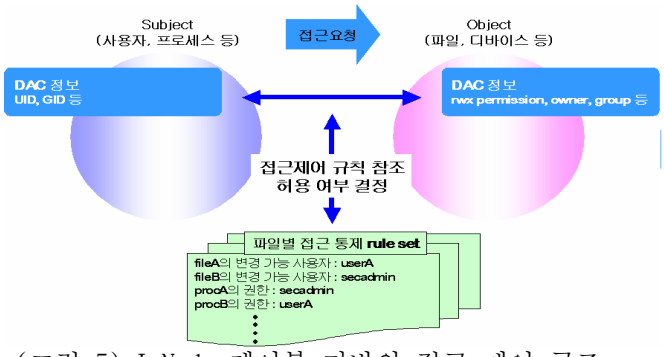
ACL 테이블 구조는 메모리 혹은 파일에 각각의 파일과 이에 대한 사용자의 허용 및 거부 정보를 테이블 형식으로 정의한 뒤 파일에 대한 접근시도가 발생할 경우 정의된 테이블에서 해당 파일의 강제적 접근제어 대상여부를 확인하고, 접근한 사용자가 해당 파일에 대하여 접근권한이 있는지 확인하여 접근 허용 및 차단 여부를 결정하게 된다.



(그림 4) ACL 테이블 구조 기반의 접근제어 구조

3.3 I-Node 레이블 구조 기반의 접근제어

I-Node 는 유닉스 시스템에서 파일 Open 시 참조되는 테이블이다. 이 테이블에는 각각 파일의 크기, 권한, 소유자, 그룹 등의 다양한 정보가 기술되어 있다. 파일에 대한 접근시도가 발생할 경우 I-Node 에서 해당 파일의 등급과 범주를 확인하고 접근 주체의 등급과 범주를 비교하여 접근 여부를 결정한다.



(그림 5) I-Node 레이블 기반의 접근 제어 구조

4. 성능 시험

[표 1] 접근제어 시험용 웹서버 환경

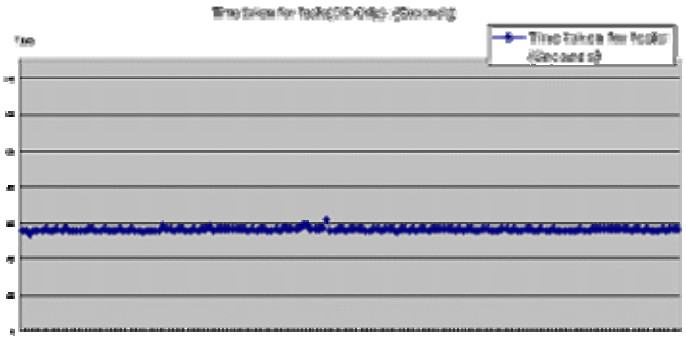
CPU	IBM PC Pentium III 750MHz Dual CPU
RAM	1024MB
HDD	20GB
OS	Solaris 8 x86

[표 2] 시험용 웹 클라이언트 환경

CPU	IBM Pentium IV 4.2GHz
RAM	512MB
HDD	40GB
OS	Windows 2000 Professional SP4

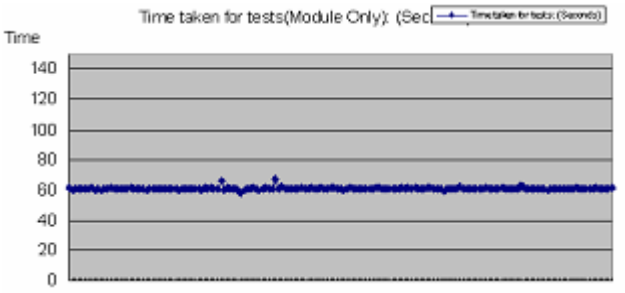
본 논문에서는 ①운영체제 보안 제품을 설치하지 않은 Solaris ②본 논문에서 구현한 ACL 테이블 구조 기반의 접근 제어 시스템 ③본 논문에서 구현한 파일시스템 레이블 기반의 접근 제어 시스템을 비교 평가하였다. [표 1]과 같은 동일한 환경의 시험용 Solaris 서버에서 성능평가를 실시하였으며, 시험 방법은 aa.html 부터 zz.html 까지 총 676 개의 html 파일을 저장하여 Apache 웹서버 2.x 를 통하여 서비스하도록 하였고, Apache 웹서버의 성능 측정툴인 Apache Bench(이하 ab)를 [표 2]와 같은 환경의 클라이언트에 설치하여 동시에 1000 세션을 처리하도록 한 후 그 처리시간을 측정하여 평균값을 구하였다. 이 실험의 목적은 접근 제어 대상이 많고 적음에 따라 시스템에 주는 영향을 비교 분석 및 시스템 부하의 선형성 여부를 확인하는데 있다. 각각의 테스트에 대한 결과를 그래프로 표현하였다.

(그림 6)에서는 OS 에 Apache 웹서버만 설치되어있는 상태에서 ab 를 이용하여 접근제어 대상 파일들을 순차적으로 Open, Close 하도록 요청하고 이를 처리하여 응답하는 시간을 누적한 그래프이다. OS 의 DAC 만 적용된 상태에서는 평균적으로 56 초가 소요되었다. 이 결과를 기반으로 ACL 방식과 I-Node 방식의 결과값을 비교해보기로 한다.



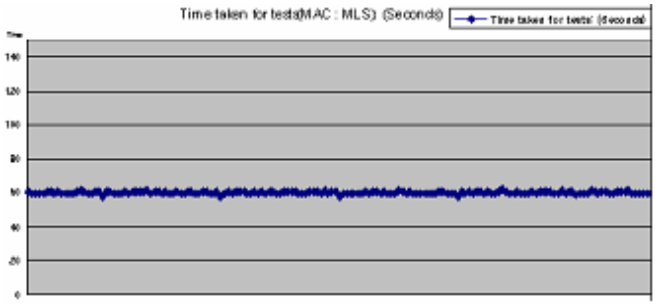
(그림 6) OS 와 Apache 웹서버에서 테스트 결과

(그림 7)에서는 OS 에 Apache 웹서버를 설치한 다음 강제적 접근제어의 후킹 모듈을 적용하되 강제적 접근제어가 적용되지 않은 상태에서의 실험결과 평균적으로 61 초가 소요되었다. 후킹 모듈을 시스템에 추가한 후 5% 정도의 오버헤드가 발생하였다.



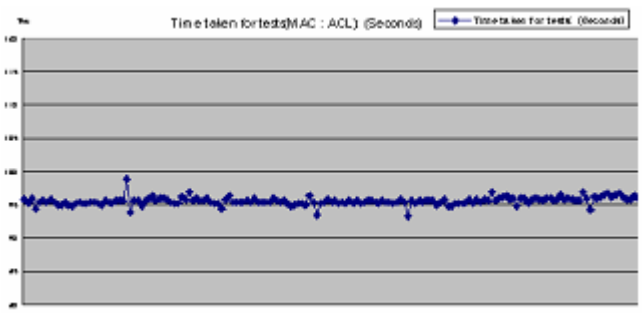
(그림 7) 강제적 접근제어를 탑재후 테스트 결과

(그림 8)에서는 OS 에 Apache 웹서버를 설치한 다음 서버에 강제적 접근제어의 후킹 모듈을 탑재하고, 각각의 접근제어 대상 html 파일의 I-Node 에 접근 제어 속성을 설정한 상태에서의 실험결과 평균적으로 60 초가 소요되었다. 이 실험 결과 시스템에 단순히 강제적 접근제어 모듈이 올라간 상태와 거의 같은 응답시간을 보여준다. 이 경우 접근제어 대상이 많고 적음과는 관련이 없음을 확인할 수 있다.



(그림 8) I-Node 레이블 테스트 결과

(그림 9)에서는 OS 에 Apache 웹서버를 설치한 다음 서버에 강제적 접근제어의 후킹 모듈을 탑재하고, ACL 테이블에 676 개의 html 파일에 대하여 강제적 접근제어를 등록한 상태에서의 실험결과 평균적으로 95 초가 소요되었다. 후킹 모듈이 시스템에 추가된 후 대략 70% 이상의 시스템 오버헤드가 발생하는 것을 확인할 수 있었고, 이에 접근제어 대상 수를 변경하면서 그 결과를 확인해 보았다.

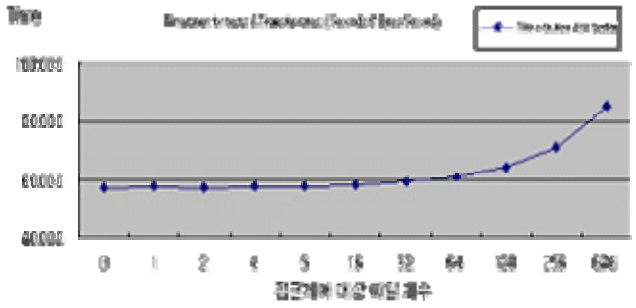


(그림 9) ACL 테이블 테스트 결과

(그림 10)에서는 접근제어 대상을 $2^n(0 \leq n \leq 9)$개로 가변하며 각각의 응답 시간을 기록한 그림이다. 접근제어 대상 파일이 16 개를 넘어서면서 테스트 응답시간이 점차 늘어나고, 대상 파일이 676 개일 때에는 93 초로 증가됨을 확인할 수 있다. 선형성을 보여야 하는 접근제어 방식이 비선형성을 보이고 있다. 결과적으로 ACL 테이블 구조의 강제적 접근제어는 접근 제어 대상이 많을수록 그 응답시간이 기하급수적으로 늘어날 수 있음을 추측할 수 있다.

[표 3] 제어 대상 파일 수와 응답시간 및 전송률

제어대상 수	응답시간(Sec)	네트워크 전송률
0	56.856	139.810
1	57.462	138.340
2	57.159	139.075
4	57.633	137.928
16	58.207	136.707
32	59.226	134.577
64	61.033	130.471
128	64.597	123.113
256	71.245	111.676
676	85.088	93.506



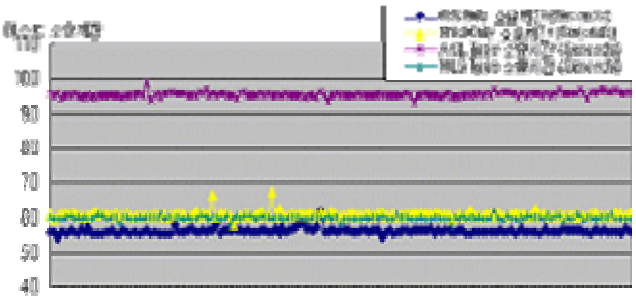
(그림 10) ACL 접근제어 파일 수에 따른 응답 비교

4. 결론

앞서 실험한 각각의 경우에 대한 테스트 결과를 종합적으로 살펴보면 (그림 11)과 같다. I-Node 를 이용한 접근제어방식과 비교하여 볼 때, ACL 테이블을 이용한 접근제어 방식은 접근제어 대상이 많으면 많을수록 시스템의 성능에 큰 영향을 주는 것을 확인할 수 있었다.

결론적으로 주체와 객체의 강제적 접근제어는 객체에 직접 접근제어 속성을 기록하고, 이를 능동적으로 적용하는 강제적 접근제어를 구현함이 바람직하다고 볼 수 있다. 현재 구현된 강제적 접근제어는 속도의 향상을 위해 I-Node 를 이용하였다.

I-Node 를 이용하는 방식은 실제 적용과정에서 해결해야 할 문제가 있다. 예를 들어, 다른 응용 프로그램에서 같은 영역을 사용하는 경우가 발생할 수 있으며 이렇게 되면 상호 간섭 등의 문제 등으로 시스템의 안정에 영향을 줄 수도 있다. 일부 OS 에서는 I-Node 영역을 사용하지 못하도록 제어하는 문제점도 발생하고 있다.



(그림 11) 접근제어방식에 따른 결과 비교

현재 미국 Sun Microsystems 사의 Solaris 의 경우 강제적 접근제어가 구현된 Trusted Solaris 와 같은 별도의 OS 가 판매되고 있다. 이는 특정 OS 에 한정된 것으로 타 상용 OS 에는 적용할 수 없다. 이를 위해 공통된 표준을 마련하고 그를 바탕으로 안정적이면서 시스템 성능에 영향을 미치지 않는 강력한 강제적 접근제어 솔루션을 구축해야 한다.

참고문헌

[1] D. D. Downs etal., "Issue in Discretionary Access Control," Proceedings of IEEE Symposium on Security and Privacy, Addison-Wesly 1998.

[2] http://www.cesg.gov.uk/site/iacs/index.cfm?menuSelected=1&displayPage=1 - BLP Access Control Model : Roos Lindgreen, Herschberg I. S., "On the Validity of the Bell-Lapadula Model," Computer & Security. Vol.13, pp.317 -338, 1994.

[3] Bell, David Elliott and Leonard J. La Padula, "Secure Computer System : Unified exposition and multics interpretation," MITRE Technical Report 2997, MITRE corp, Bedford, MA, 1975.

[4] 분산통신망 환경에서의 통합 접근통제 시스템 연구, 연구수행기관: 대전대학교, 1998.

[5] 김현정, 박태규, 조인구, 임연호, “다중 등급 보안 리눅스 시스템 개발과 보안 인터페이스,” 티에스온넷, 한서대학교, 정보보호학술발표회논문집, 2000.