안전한 운영체제 기술개발 동향

김재명, 이규호, 김종섭, 김귀남
경기대학교 정보보호기술공학과

요 약

전 세계가 정보화의 물결 속에서 보다 신속하고 안전한 정보의 교류를 위한 기술 개발 및 연구는 시간의 변화와 더불어 급변하고 있다. 정보의 중요성과 효용가치의 증대는 다양한 분야에서의 기술 발전을 함께 필요로 하게 되었고 특히, 정보보호의 필요성은 그 핵심기술로 자리잡게 되었다. 하지만, 대부분의 시스템 공격이 운영체제의 취약성을 기반으로 하고 있어, 응용 레벨에서의 보안 기술만으로는 소기의 목적을 달성하기에 어려움이 있다. 이런 문제점을 해결하기 위해 운영체제 자체에서 안전하고 신뢰성 있는 서비스를 제공하는 기술에 대한 연구가 많이 이루어지고 있다. 따라서 본 고에서는 현재 보안커널의 세계적인 추세와 동향을 파악하고자 안전한 운영체제 개발 프로젝트들을 소개하고, 전자서명 검증을 운영체제 수준에서 수행하여 홈페이지 등의 파일시스템의 위·변조를 원천적으로 방지하는 File Griffin에 구현된 보안 메커니즘에 대하여 알아본다.

Secure OS Technical Development Trend

Jai-myong Kim, Kyu-ho Lee, Jong-seob Kim, Kuinam J Kim

ABSTRACT

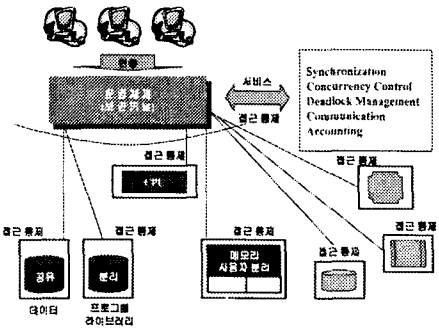
In the 3rd Wave of information revolution, technical research & development for more rapid and safe information exchange take a sudden turn currently. According to a step up in importance and efficiency value of information, it's necessary to research technical development in various field altogether. Especially information security is the very core of essential technology. However most System attacks are based on the weakness of OS, it is difficult to achieve the security goal in the only application level. For the solution of this problem, so many technology researches to serve secure, trust information security in OS itself are activated. Consequently we introduce the tendency of current secure OS development projects of security kernel all over world in this report and inquire into security mechanism of the File Griffin which prevents file system forgery, modification perfectly by performing digital signature certificate on kernel level.

1. 서 론

안전한 운영체제(Secure Operating System)란 컴퓨터 운영체제상에 내재된 보안상의 결함으로 인하여 발생 가능한 각종 해킹으로부터 시스템을 보호하기 위하여 기존의 운영체제 내에 보안기능을 통합시킨 보안커널(Security Kernel)을 추가로 이식한 운영체제이다[1]. 보안커널이 이식된 운영체제는 컴퓨터 사용자에 대한 식별 및 인증, 강제적 접근통제(MAC : Mandatory Access Control), 임의적 접근통제(DAC : Discretionary Access Control), 재사용 방지(Object Reuse Prevention), 침입 탐지(Intrusion Detection) 등의 보안 기능 요소를 갖추어야 한다([그림 1] 참조) [2]

인터넷과 같은 네트워크 환경에서 유닉스가 가지는 "개방성"은 중요한 특성이지만 컴퓨터 내의 정보보호를 향상시키기 위한 도구는 현재의 표준 유닉스에서는 매우 부족한 실정이다. 한 통계자료에 의하면 성공한 네트워크 공격의 약 8%가 유닉스 시스템 자체의 취약점을 공격함으로 해서 이루어졌다고 한다. 이에, 기존 유닉스 시스템의 취약점을 보완하는 패치나 업그레이드를 통한 임시방편적인 방법보다는 원천적으로 새로운 안전한 유닉스 운영체제의 필요성이 대두되고 있다.

본 고에서는 안전한 운영체제와 보안커널 개념을 소개하고, 오픈 프로젝트인 DTOS, Fluke/Flask, RSBAC(Rule Set Based Access Control) 프로젝트를 소개하고, 운영체제 수준에서 전자서명인증 메커니즘을 통한 신원확인을 통하여 홈페이지 등의 파일시스템에 대한 강력한 접근통제를 수행하여 원천적으로 해킹을 방지하는 File Griffin에 구현된 보안 메커니즘에 대하여 설명한다.

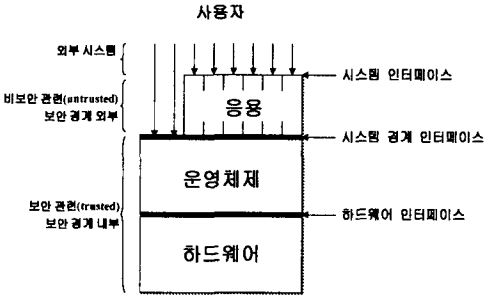


[그림 1] 운영체제 보안기능

2. 안전한 운영체제 및 보안커널

2.1 컴퓨터 시스템의 구조

전통적인 컴퓨터 시스템의 구조는 [그림 2]와 같이 하드웨어, 운영체제 및 응용프로그램으로 구성된다. 그림에서 각각의 계층은 아래 계층에 있는 facility를 사용한다. 운영체제와 하드웨어는 보안관련으로 보안경계(Security Perimeter) 내부에 위치한다. 응용프로그램은 잘 정의된 시스템 콜을 사용하여 보안경계를 통하여 운영체제에 접근한다. 사용자들은 시스템 외부에 있으며, 운영체제와 직접 통신하거나 응용프로그램을 통하여 시스템에 접근한다.



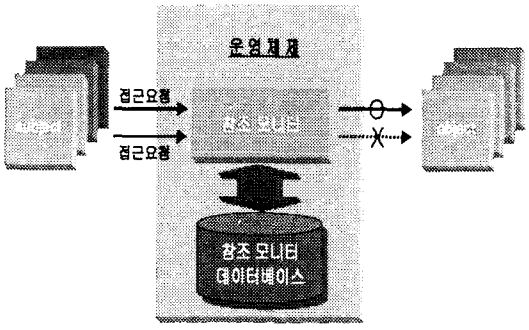
[그림 2] 일반 컴퓨터 시스템 구조

2.2 보안커널의 개념

보안커널은 전통적으로 미국 등의 선진국으로부터 개념이 정립되어 왔으며, 다양하게 정의를 내리고 있지만, 대체로 다음과 같이 정리될 수 있다.

- "참조모니터(Reference Monitor)의 개념을 정의한 TCB(Trusted Computing Base)의 하드웨어, 펌웨어, 소프트웨어 요소 [DoD]" [2]
- "보안커널이란 시스템 자원에 대한 접근을 통제하기 위한 기본적인 보안절차를 구현한 컴퓨터의 중심부 [FIPS]"

참조모니터란 보안커널의 가장 중요한 부분으로 객체에 대한 접근통제기능을 수행하고 감사, 식별 및 인증, 보안 매개변수 설정 등과 같은 다른 보안메커니즘과 데이터를 교환하면서 상호작용을 한다([그림 3] 참조). 참조모니터는 항상 호출되어야 하고, 참조모니터에 대한 부정행위는 방지되어야 하며, 분석과 시험이 용이하도록 충분히 작아야되는 요구사항을 가지고 있다.



[그림 3] 참조모니터의 개념

TCB는 보안정책의 시행을 책임지는 하드웨어, 펌웨어, 소프트웨어 및 이들의 조합을 포함하는 컴퓨터 시스템 내의 모든 보호 메커니즘으로서 기본적인 보호환경을 제공하며, 운영체제의 기본적인 작업(프로세스 활성화, 실행영역 교환, 메모리 보호, I/O 연산 등)에 대한 보안성 및 무결성을 감시하는 기능을 수행한다. [3][4]

2.3 보안커널의 설계원리

보안커널은 기존의 운영체제에 내재된 보안상의 결함으로 인한 각종 침해로부터 시스템을 보호하기 위하여 기존의 운영체제 내에 추가적으로 이식되어, 사용자의 모든 접근 행위가 안전하게 통제되게 한다.

보안커널은 일반적으로 운영체제와 유사하며, 전통적인 운영체제 설계 개념을 사용한다. 보안커널에 요구되는 하드웨어도 거의 유사이다. 보안커널은 보안경계 내의 모든 주체와 객체를 통제하여야 하며 프로세스, 파일시스템, 메모리 관리, I/O를 위한 자원을 제공하여야 한다. 일반적으로 보안커널의 개념을 운영체제에 구현하기 위해서는 [표 1]과 같은 설계원리가 적용되고 있다. [6]

보안커널 설계 시 커널의 구조 설계와 함께 운영체제가 지원해야 할 보안기능에 대해서도 고려를 해야 하는데, 운영체제는 최소한 [표 2]의 보안기능을 제공하여야 한다.

[표 2] 보안커널의 설계원리

특 징	내 용
최소권한	• 사용자와 프로그램을 가능한 최소의 권한으로 운영하여, 우연 혹은 의도적인 공격으로부터 손상을 최소화
보호메커니즘의 경제성	• 충분한 분석 및 검증이 가능하도록, 작고 단순한 보안 시스템 설계
개방형 설계	• 충분한 검토가 가능하도록 상대적으로 작고 주요한 보안메커니즘에 의존하고 공개
완전한 중재 및 조정	• 직접적 혹은 우회적인 모든 접근에 대한 검사
허용에 기반한 접근	• 객체에 대한 접근은 거부가 기본
권한 분리	• 객체에 대한 접근은 하나 이상의 조건에 의하여 결정되어, 하나를 우회한다 하더라도 객체의 보호가 이루어져야 함
최소공통 메커니즘	• 공유객체는 정보흐름의 가능성이 있는 채널을 제공하므로 이를 최소화
사용의 용이성	• 보안메커니즘은 사용이 용이하여 우회가능성이 적어야 함

보안커널 설계 시 커널의 구조 설계와 함께 운영체제가 지원해야 할 보안기능에 대해서도 고려를 해야 하는데, 운영체제는 최소한 [표 2]의 보안기능을 제공하여야 한다.

[표 3] 보안커널의 보안기능

기 능	설 명
식별 및 인증	• 고유한 사용자 신분에 대한 인증 및 검증
강제적 접근통제	• 사용자의 접근결정에 사용되는 고정된 보안속성을 보안관리자 또는 운영체제에 의해 정해진 엄격한 규칙에 따라 자동적으로 부여함으로써 사용자의 자유재량에 상관없이 강제적으로 접근통제
임의적 접근통제	• 사전에 보안정책이나 보안관리자에 의해 개별 사용자에게 합법적으로 부여한 한도 내의 재량권에 따라 사용자가 그 재량권을 적용하여 접근통제
객체 재사용 방지	• 메모리에 이전 사용자가 사용하던 정보가 남아 있지 않도록 기억장치 공간을 깨끗이 정리
완전한 중재 및 조정	• 모든 접근경로에 대한 완전한 통제
감사 및 감사기록 축소	• 보안관련 사건 기록의 유지 및 감사기록의 보호 • 막대한 양의 감사기록에 대한 분석 및 축소
안전한 경로	• 패스워드 설정 및 접근 허용의 변경 등과 같은 보안관련 작업을 수행할 때 안전한 경로 제공
침입탐지	• 정상적인 시스템의 사용 패턴을 분석하고, 비정상적인 사용이 발생했을 때 이에 대한 경보 제공

3. 국내·외 보안커널 오픈 프로젝트 동향

3.1 Synergy - DTOS

1) Synergy 프로젝트

미국 정부에서는 정부기관 및 군사 기관들의 컴퓨터 네트워크에 사용할 목적으로 안전한 유닉스 운영체제 개발을 사업화하여 수년 전부터 타당성 검토를 마치고 현재 프로토타입이 개발된 상태에 있다.

NSA(National Security Agency)에서 NIST(National Institute of Standards and Technology), DISA(Defence Information Security Agency), ARPA(Advanced Research Projects Agency) 등의 기관과 공동으로 추진중인 Synergy 연구 프로그램(Synergy research program)은 미국의 "High Performance Computing and Communications (HPCC) : Advancing the Frontiers of Information Technology" 프로그램 중의 하나이다. 1993년 11월 23일 시행령 12881에 의해 클링턴 정부가 만든 NSTC(National Science and Technology Council)의 CCIC(Committee on Computing, Information, and Communications)의 일부인 HPCC는 정보기술, 컴퓨팅, 통신 및 정보기반구조를 이끌어 21세기에도 자국 국민들의 생산성을 유지할 수 있도록 경쟁력을 유지하고 확장시키기 위한 것이다[5].

2) Synergy 설계 원칙

Synergy 연구 프로그램의 설계 원칙은 [표 3]과 같다

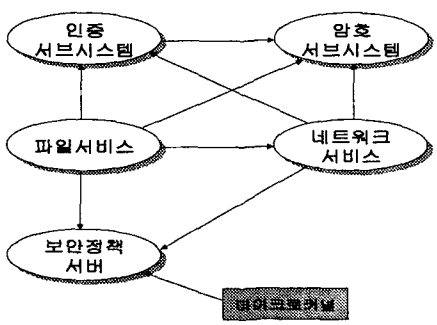
[표 4] Synergy 연구 프로그램의 설계 원칙

원 칙	내 용
유연성(Flexibility)	• 위협 환경에 따라 구성가능한 보안 서비스 • 메커니즘에 독립적인 보안 서비스 • 일반 보안 API 내에 캡슐화된 보안 서비스/메커니즘
투명성(Transparency)	• 운영체제 기반구조에 통합된 보안 • 자동적으로 시작되는 보안 서비스 • 자동화된 보안관리 기능
모듈성(Modularity)	• 독립적으로 평가될 수 있는 모듈로 분해된 구조 • 모듈간에 잘정의된 인터페이스 및 상호독립성 • 모듈의 구성에 대해 판단할 수 있는 능력
계층적 보안 (Layered Security)	• 가능한 많은 위협을 처리할 수 있는 기본 시스템 구성요소 • 다른 TCB 구성요소상에 최소한으로 증명된 의무

3) Synergy 구조

Synergy의 구조는 [그림 4]와 같이 마이크로커널(Microkernel) 기반구조이며, 보안정책 서버(Security policy server), 암호 서브시스템(Cryptographic Subsystem), 인증 서브시스템(Authentication Subsystem)으로 구성된 보안서비스 서브시스템으로 구성되어 있으며, 구성

가능한 네트워크 및 파일 서비스 구조를 가지고 있다.



[그림 4] Synergy 구조

마이크로커널 기반구조를 사용하는 이유는 위에서 언급한 Synergy 설계원칙을 만족시키기 위한 것이다. 메커니즘과 정책간의 분리를 강조함으로써 유연성을 제공할 수 있다. 또한 마이크로커널에 의해 많은 위협을 투명하게 처리할 수 있다. 또한, orthogonal 서버로 운영체제를 분해함으로서 모듈성을 제공하며, 마이크로커널이 운영체제 서버의 활동들을 제한함으로서 계층화된 보안을 제공할 수 있다.

보안정책 서버는 강제적 접근통제(MAC, Mandatory Access Control)[1][2]를 제공하기 위한 것이며, MAC의 세부 항목(마이크로커널, 파일 서비스, 네트워크 서비스, 응용)으로부터 정책 시행자를 분리시킴으로서 MAC 정책이 변경되었을 경우, 변경된 부분만을 재평가함으로써 비용을 줄일 수 있다.

암호 서브시스템은 암호기능을 제공하기 위한 것으로 암호사용 정책과 암호메커니즘에 대한 내용을 다룬다. 암호알고리즘의 세부 항목(파일 서비스, 네트워크 서비스, 응용)으로부터 클라이언트를 분리시킴으로서 암호 기능의 변경시에 비용을 줄일 수 있다.

4) DTOS

DTOS(Distributed Trusted Operating System)는 NSA의 Synergy 프로그램 중의 일부로 강력하고 유연성 있는 보안 통제 제공을 목적으로 만든 운영체제의 프로토타입으로 1997년 6월에 개발·완료되었다 [7].

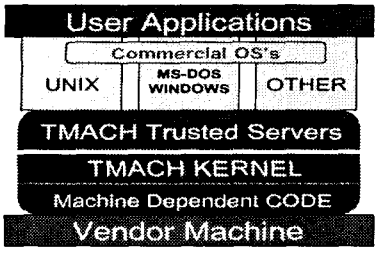
현재의 Synergy 노력은 차세대에 상용 운영체제에 강력한 보안 메커니즘을 포함하도록 운영체제 개발자들을 독려하는 장기적인 전략중의 하나이다.

DTOS 프로그램의 주목적은 Synergy 프로그램의 기본적인 요구사항을 만족시키기 위한 것으로 Synergy 구조내에서의 최하위수준의 소프트웨어 구성요소에 대한 프로토타입을 개발하고, 보안 요구사항을 만족시키기 위한 보증 문서와 증거의 제공 즉, Synergy 구조를 따르는 전체 시스템의 보안에 대한 증거를 어떻게 다룰 것인가를 연구하는 것이다.

DTOS는 High assurance multilevel secure 환경에 사용하기 위해 B3 등급을 목표로 설계된 DTMach 프로그램[그림 5]을 대체하는 것으로 안전한 분산 운영체제를 위한 고수준의 설계를 하는 것이다. DTOS 프로토타입 개발은 거의 마이크로커널과 보안서버의 설계로 집중된다.

마이크로커널은 기존의 Mach3.0을 개선하는 방향으로 설계되었다. 이는 기존의 마이크로커널에서 요구하는 보안에 대한 필요성이 크게 다르지 않다는 것을 나타낸다. Mach 마이크로커널을 선택한 이유는 하나의 통제 메커니즘으로 거의 모든 시스템 오퍼레이션을 통제할 수 있으며, 널리 사용되는 플랫폼이라는 점과 함께 유닉스 응용을 지원하며, 소스코드를 별도의 동의 절차 없이 사용할 수 있었던 등등 다양하다.

보안서버에는 시스템의 보안정책이 포함되는데 보안정책의 주 기능은 마이크로커널과 같은 다른 서비스를 제공하는 구성요소에 의해 시행되는 보안결정(security decision)을 제공하는 것이다.



[그림 5] DTMach 구조

3.2 Fluke / Flask

1) 개 요

Synergy-DTOS 과제의 결과물로 Mach 마이크로커널 프로토타입이 나왔으나 이 버전의 Mach에 대한 연구는 지속되지 못했다. 이에 NSA, Utah 대학, SCC(Secure Computing Corp.)에서 공동으로 Utah 대학에서 개발중이던 Fluke 운영체제의 보안 기능을 강화한 운영체제 개발 프로젝트를 진행하였다. 이 프로젝트의 목적은 DTOS의 보안 구조를 기반으로 다양한 보안 정책을 수용할 수 있는 유연성(Flexibility) 지원과 중요한 보안 기능을 정형화된 방법으로 검증이 가능하도록 하는 것이다. [8]

2) 보안 구조

Flask의 보안 구조는 다음과 같은 목적으로 가진다.

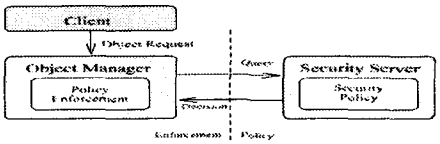
- 보안정책 유연성
- 응용 프로그램에 대한 투명성 (Application transparency)
- 높은 보안 수준 (Defense-in-depth)
- 보증의 용이성 (Ease of assurance)
- 정책 변화에 따른 최소한의 코드 변환

이러한 보안 목적을 달성하기 위해서는 객체에 대한 모든 접근 중재가 가능하도록 주체와 객체간의 분리 기능이 존재해야 하며, 객체에 대한 접근을 시도하는 주체에 대한 안전한 신분 확인 기능이 제공되어야 한다.

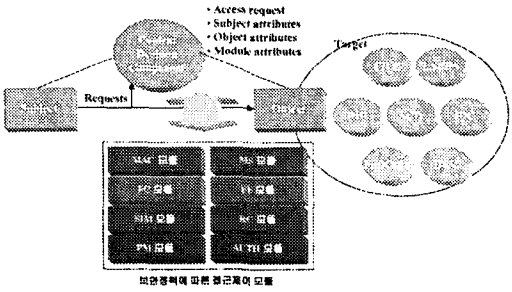
3) 구성 요소

Flask의 구성요소는 다음과 같다. 보안 정책을 시행하는 부분을 객체 관리자라 하며, 수립된 보안 정책에 따라 보안 결정을 내리는 부분을 보안 서버라 한다. 보안 서버의 정책 다양한 정책이 존재할 수 있으며, 이들은 시스템 구조에 의존적이지 않다. (그림[10] 참조)

- 객체 관리자 (Object Manager) : 시스템 제어 동작을 제공하는 서버들로 구성
  - 가상메모리 관리자
  - 파일 서버
  - 프로세스 관리자
  - 네트워크 서버
- 보안 서버 (Security Server) : 특정 보안 정책에 대한 보안 결정을 담당하며 다음의 기능을 수행
  - 보안 정책 결정
  - SID(Security ID)와 보안 문맥 (Security context)간의 매핑
  - 객체 관리자 AVC(Access Vector Cache)관리 및 정책 적용
  - 네 가지 보안정책(MLS : Multi-Level Security, TE : Type Enforcement, 신분기반 접근통제, 동적 역할기반 접근통제)의 조합으로 정책 수행
- 마이크로 커널 (Micro Kernel) : Fluke Micro Kernel



[그림 6] Flask 구조



[그림 7] RSBAC 보안 메커니즘

3.3 RSBAC (Rule Set Based Access Control for LINUX)

1) 개 요

RSBAC은 독일의 Hamburg 대학에서 수행중인 리눅스 커널 기반 보안 기능 확장 프로젝트로 Abrams & Lapadula에 의해 제안된 GFAC(Generalized Framework for Access Control) 기반 접근제어 모델의 사용으로 다양한 보안정책 적용이 가능하다. 보안 기능을 수행하는 system call을 추가하여 보안 기능을 제공한다.

모든 결정은 접근 주체(Subject)와 그 속성, 접근 객체(Object)와 그 속성, 그리고 접근 타입에 따라 결정되며, 이 결정에 필요한 모든 정보는 각각 결정 모듈에 독립적으로 마운트(mount)된 장치에 저장되어 이 정보에 접근하기 위해 특정한 보안 기능을 갖춘 시스템이 제공된다. 현재 안정버전 1.1.1까지 개발된 상태이다. [9]

2) RSBAC 보안 메커니즘 및 보안정책 모듈

RSBAC은 [그림 7]과 같은 보안 메커니즘을 가지며, 각 보안정책 모듈은 [표 4]와 같다.

[표 5] RSBAC 보안정책 모듈

보안정책 모듈	기 능
MAC (Bell-Lapadula Mandatory Access Control)	• 주체 및 객체의 보안등급에 따라 접근 허용 여부를 결정하는 접근제어 모델로 높은 보안등급의 정보가 낮은 등급으로 흐르는 것을 방지
FC (Functional Control),	• 사용자에게는 역할(role), 객체에게는 카테고리(category)를 부여하여 이에 따라 접근 결정을 내리는 단순화된 RBAC(Role-based Access Contorl) 모델
SIM (Security Information Modification)	• 오직 Security Officer만이 보안정보에 대해 변경, 쓰기 가능
MS (Malware Scan)	• 바이러스에 걸린 파일의 실행과 같은 시스템 파괴 동작을 감지 • 스캐너 알고리즘에 의해 알려진 바이러스만이 탐지 가능
FF (File Flags)	• 파일과 디렉토리에 접근 플래그를 정의하여 특정 역할을 가진 사용자에게만 접근 허용 • Security Officer 역할을 가진 사용자만이 플래그 변경 가능
RC (Role Compatibility)	• 사용자에게는 64종류의 역할(role)을 부여하고, 각 객체에게는 64종류의 타입을 부여하여 각 역할과 객체 사이의 호환성 있는 관계를 설정하여 융통성 있는 접근제어 규칙 생성
AUTH (Authentication)	• 실행하려는 대상 프로세스/프로그램의 소유자(setuid)를 임의로 바꾸는 것을 인증된 사용자에게만 제한

4. File Griffin - SECUVE

1) 개 요

국내에서 상용화된 보안커널 제품군으로는 Secuve의 File Griffin을 들 수 있다. 서버 보안 커널 제품으로 홈페이지의 등의 파일시스템에 대한 불법적인 해킹을 방지하기 위해 특성화 및 최적화된 보안커널이며, UNIX 기본 권한 외에 관리자 및 사용자를 커널에서 전자서명 메커니즘을 이용하여 인증함으로써, 기존의 운영체제 취약성을 이용한 공격을 통해 시스템 관리자(root) 권한을 획득할지라도, 시스템 자원(파일)에 접근할 수 없어 파일 시스템의 무결성 및 가용성을 보장해 준다.

File Griffin이 제공하는 주요 기능으로는 전자서명 인증방식의 사용자 신원확인으로 기본의 응용계층이 아닌 커널수준에서 지원되며, 이는 사용자 신원확인에 대한 원천적인 안전·신뢰성을 제공한다. 또한, File Griffin은 기존 UNIX가 제공하는 파일에 대한 접근권한 비교 이전에 전자서명 인증의 접근권한과 파일에 설정되어 있는 접근권한을 비교하는 권한판정 메커니즘이 먼저 적용된다. 이는 UNIX의 맹점이었던 시스템관리자(root)라 할지라도 해당파일에 File Griffin의 접근권한이 주어지지 않으면 접근하지 못함을 의미한다.

File Griffin은 전자서명 인증 및 파일 접근통제외에도 주요 프로세스에 대한 불법종료 및 시스템 불법정지 방지 기능을 제공한다. 이는 해커가 주요 프로세스 및 시스템을 비정상적으로 종료하는 것을 방지함으로써 시스템이 제공하는 서비스의 가용성(Service Availability)을 보장해 준다.

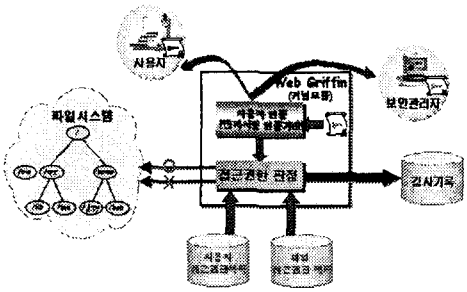
[표 6] File Griffin 주요 특징

특징	설 명
전자서명인증기반의 사용자 신원확인	• 시스템 사용자 신원확인을 위한 국제표준인 RSA 전자서명 인증 방식을 이용하여 인증서와 서명문을 커널 레벨에서 검증
보안커널에 의한 강력한 파일시스템 접근통제	• 커널 수준에서 참조 모니터(Reference Monitor)에 기반한 강력한 접근통제 기능 수행
Kernel Sealing에 기법에 의한 강력한 보안커널 구축	• 커널실링(Kernel Sealing)에 의한 커널 동적 모듈 적재 및 삭제 등에 대한 사용제한 기능제공
웹데몬 및 중요 데몬 프로그램의 불법종료 방지	• 웹 데몬 등의 주요 데몬 프로그램의 불법 종료 방지 기능 제공
컴퓨터 시스템의 불법정지 방지기능 제공	• 시스템 정지 명령어(reboot 등) 사용의 제한 기능 지원 • 시스템 불법정지 방지로 인한 시스템 운영의 가용성 보장
커널레벨의 보안감사 제공	• 보안관리자 및 사용자의 모든 보안관련 이벤트를 커널에서 제공
원격클라이언트 및 이기종 통합 관제기능 지원	• 원격 접속을 위한 전자서명 인증기반의 전용 클라이언트 제공 • 분산환경에서 운영되는 이기종 플랫폼의 통합관리 기능 제공

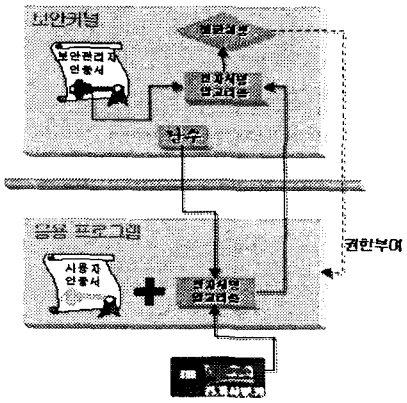
2) File Griffin의 구조

File Griffin의 구조는 [그림 8]와 같다.

- 사용자 인증 모듈 : PKI(Public-Key Infrastructure) 기반의 인증서를 통한 인증 과정을 통해 사용자에게 접근 권한을 할당해 준다.
- 접근권한 판정 모듈 : 파일 시스템에 접근하는 시스템 콜을 가로채어(hooking), 사용자의 권한과 사용자 ACL(Access Control List), 파일 ACL을 참조하여 접근 주체(사용자 or 프로세스)의 접근 여부를 결정하고, 해당 사건에 대한 감사기록(Audit Record)를 생성한다.
- ACL Database : File Griffin은 시스템 자원에 대한 접근권한을 판단하기 위하여 사용자와 관련된 접근통제목록(User ACLs)과 파일에 관련된 접근통제목록(File ACLs)



[그림 8] WebGriffin 구조



[그림 9] 커널에서의 전자서명 인증 과정

3) 커널 내부의 전자서명 인증 메커니즘

커널 내부의 전자서명 인증 과정은 [그림 9]와 같다. File Griffin이 지원하는 전자서명 알고리즘은 RSA 및 DSA이며 전자서명키의 키길이는 1,024비트이다. 또한 지원하는 해쉬 알고리즘은 MD5, SHA-1 알고리즘이다.

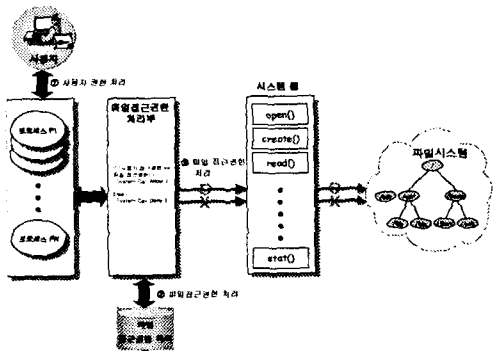
- ◆사용자가 인증을 요청하면, 커널에서 난수(Random number)를 생성하여 이를 사용자에게 넘겨준다.
- ◆사용자는 난수에 자신의 비밀키로 전자서명을 한 후 서명값과 인증서를 커널에 넘겨준다.
- ◆커널은 서명값과 인증서를 가지고 인증서 검증 작업을 수행하여, 올바른 인증서임이 확인될 경우 사용자를 인증하며, 사용자의 프로세스에게 접근권한을 할당해 준다.
- ◆이 후, 사용자가 시스템 자원에 접근하려 할 때, 할당된 접근 권한과 시스템 자원의 권한 속성을 가지고 접근권한 판정을 하게 된다.

4) 커널 수준의 접근통제 메커니즘

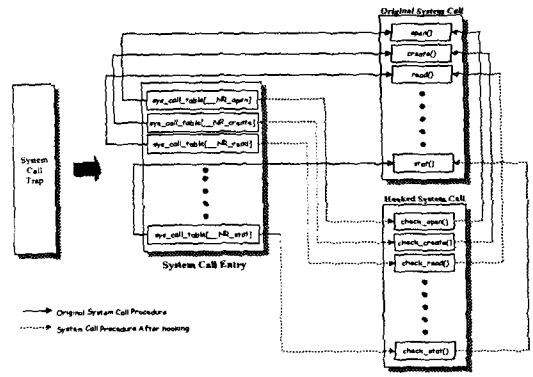
Web Griffin의 파일에 대한 접근통제 처리메커니즘은 [그림 10]과 같으며, 처리 절차는 다음과 같다.

① 사용자 접근권한 처리 : 전자서명 인증(2.장 참조)을 거친 사용자에 대해 보안커널은 사용자의 접근권한을 프로세스에게 할당하고, 필요시 사용자의 프로세스로부터 접근권한을 읽어온다.

② 파일 접근권한 처리 : 보안커널은 권한이 설정된 파일에 대한 접근권한을 파일 접근권한목록으로부터 읽어오기도 하며, 사용자의 요구에 의하여 사용자의 접근권한으로 파일에 대한 접근권한을 설정할 수 있다.



[그림 10] Web Griffin 파일 접근통제 메커니즘



[그림 11] UNIX 시스템에서의 시스템콜 후킹(System Call Hooking)

③ 파일에 대한 접근통제 처리 : 보안커널은 프로세스의 접근권한과 파일에 대한 접근권한이 일치하면, 파일에 대한 접근 및 연산인 시스템 콜을 허용하고 그렇지 않으면, 거부한다.

Web Griffin은 파일에 대한 모든 접근을 통제하기 위하여 파일연산과 관련된 모든 시스템콜 후킹(System Call Hooking)기술을 사용하고 있다([그림 11] 참조).

시스템콜 후킹은 원래의 시스템콜 엔트리에 있던 시스템콜을 다른 곳에 위치시키고, 새로운 루틴이 추가된 모듈을 원래의 시스템콜 엔트리에 등록시키는 기술을 말한다. 이 경우 해당 시스템콜에 대한 수행여부는 후킹된 시스템콜 모듈에서 접근권한을 비교함으로써 판별한다.

시스템콜 후킹 기술은 커널 주소 공간으로 커널모듈을 로딩할 수 있는 운영체제에서 시스템 자원에 대한 접근통제 수행을 위하여 널리 쓰이는 메커니즘이다.

4. 결 론

서론에서 언급한 바와 같이, UNIX, 리눅스 등 기존의 운영체제에 기반한 보안 기능이나 응용 프로그램을 통한 안전성 실현에 한계를 있음을 인식하면서, 이에 대한 원천적인 해결책으로 보안커널에 대한 연구·개발이 국내·외적으로 여러 가지 형태로 진행되고 있음을 알 수 있었다. 이에 더불어 세계가 정보화사회로 급변해 가면서, 정보보호 기반 기술은 국력의 척도가 되어가고 있으므로, 국내에서도 보안커널에 대한 관심을 가지고, 많은 연구가 이루어져야 할 것이다.

참고문헌

[1] Charles P. Pereeger, Security In Computing, Prentice Hall, 2nd Ed., 1997.

[2] National Computer Security Center, Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, Dec., 1985.

[3] 이철원, 홍기융, 김학범, 오경희, 심주길, "다중등급 보안정책을 지원하는 침입차단시스템의 설계", 제7회 통신 정보 합동 학술 대회(JCCI '97) 논문집, pp. 59 ~ 63, 4. 1997.

[4] C. W. Lee, K. Y. Hong, K. H. Oh, H. B. Kim, J. G. Sim, "Firewall System for Multilevel Security Environment", JWISC Fall Conference Proceedings, pp. 147 ~ 152, Oct., 1997.

[5] Synergy Project http://www.cs.utah.edu/~sds/synergy/

[6] Gery Herman, "Operating Systems Security for midrange and Large Computer: Overview", DATAPRO, Dec. 12. 1996.

[7] Secure Computing Corporation, DTOS Lessons Learned Report, 27. June, 1997.

[8] Flax : Flux Advanced Security Kernel, http://www.cs.utah.edu/flux/fluke/html/flask.html

[9] Rule Set Based Accesss Control for Linux, http://www.rsbac.de/

[10] Web Griffin - Secuve, http://www.secuve.com

김 재 명

1997년 충남대학교 컴퓨터과학과(공학사)

1999년 충남대학교 컴퓨터과학과(공학석사)

2001년 경기대학교 정보보호기술공학과(박사과정)

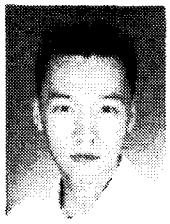
1996년 ~ 1997년 한국정자통신연구원 위촉연구원

1998년 ~ 2001년 한국정보보호진흥원 연구원

1999년 ~ 2001년 전자서명 공인인증기관 실질심사 위원

2001년 ~ 현재 (주)시큐브 상무이사

이 규 호



1999년 아주대학교 정보및컴퓨터공학부(공학사)

2001년 아주대학교 컴퓨터공학과(공학석사)

2001년 경기대학교 정보보호기술공학과(박사과정)

2001년 ~ 현재 (주)시큐브 선임연구원

김 종 섭

1978년 중앙대학교 법학과(법학사)

2000년 동국대학교 정보보호학과(정보보학석사)

2001년 경기대학교 정보보호학과(박사과정)

1996년 ~ 1999년 한국정보보호진흥원 협력관

1981년 ~ 현재 경찰청 사이버테러대응센터 경감

김 귀 남

미국 캔자스대학 수학과(응용수학사)
미국 콜로라도주립대학 통계학과(통계학석사)
미국 콜로라도주립대학 기계·산업공학과(기계·산업공학박사)
현재 경기대학교 정보보호기술공학과 주임교수