Journal of the Korean Association of Geographic Information Studies, 26(4) 2023, pp.67-79 ISSN 1226-9719(Print)
https://doi.org/10.11108/kagis.2023.26.4.067 ISSN 2287-6952(Online)

데이터 보호를 위한 파일시스템 기반의 SecureOS Module에 관한 연구*

장용구[1] · 김인철[2※] · 류지송[3]

Research on SecureOS Module Based on File System for Data Protection*

Yonggu JANG[1] · Inchul KIM[2※] · Jisong RYU[3]

요 약

노트북, 스마트 기기 및 다양한 IoT 장비를 통한 서비스 환경은 매우 빠르게 발달하고 있다. 이러한 인터넷 환경에서 최근의 보안 대책은 주로 네트워크 응용 수준의 보안 대책인 방화벽(침입 차단 시스템-Firewall)과 IDS(침입 탐지 시스템-Intrusion Detection System)으로 이루어지고 있다. 또한, 최근에는 다양한 보안 데이터의 현장 활용이 이루어지고 있고 이런 보안 데이터의 관리와 파기에 대한 이슈 소요가 제기되고 있다. 이러한 보안 데이터의 관리를 위해 문서보안(DRM:Digital Rights Management)이나 데이터 손실 방지 솔루션(DLP:Data Loss Prevention)과 같은 제품이 사용되고 있다. 그러나 이런 보안 대책에도

사용성 문제로 인해 현장에서 사용하기 위해 반출된 데이터 보안 대책은 대부분 환경에서 해당 데이터를 암호화하여 전달하고 저장하는 정도로 운영되고 있으며, 암호키 관리나 데이터의 파기에 관한 대책이 미흡한 것이 현실이다. 이러한 문제점을 기반하여 OS 기반의 보안 모듈을 제공함으로써 사용자는 동일한 인터페이스로 보안 데이터를 관리 운영할 수 있는 SecureOS Module을 제시하고자 한다.

주요어 : 보안, 데이터 보호, SecureOS, 데이터 접근 권한, 데이터 파기, 암호화

ABSTRACT

Service environments through laptops, smart devices, and various IoT devices are

2023년 10월 14일 접수 Received on October 14, 2023 / 2023년 10월 31일 수정 Revised on October 31, 2023 / 2023년 11월 06일 심사완료 Accepted on November 06, 2023
* 본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호 : RS-2020-KA158151)
1 한국건설기술연구원 연구위원 / Research Fellow, Korea Institute of Civil engineering and Building Technology
2 (주)에스큐브아이 이사 / Director of SCUBEI CO.,LTD
3 한국건설기술연구원 수석연구원 / Senior Researcher, Korea Institute of Civil engineering and Building Technology
※ Corresponding Author E-mail: mp3250@s3i.co.kr

developing very rapidly. Recent security measures in these Internet environments mainly consist of network application level solutions such as firewall(Intrusion Prevention Systems) and IDS (intrusion detection system). In addition, various security data have recently been used on-site, and issues regarding the management and destruction of such security data have been raised. Products such as DRM(Digital Rights Management) and DLP(Data Loss Prevention) are being used to manage these security data. However despite these security measures, data security measures taken out to be used in the field are operated to the extent that the data is encrypted, delivered, and stored in many environments, and measures for encryption key management or data destruction are insufficient. Based on these issues we aim to propose a SecureOS Module, an OS-based security module. With this module users can manage and operate security data through a consistent interface, addressing the problems mentioned above.

KEYWORDS : security, data protection, secureOS, data access authority, destroy data, encryption

서 론

1. 연구의 배경 및 목적

보안 데이터를 관리하기 위하여 데이터를 암호화하여 저장하고, 암호화/복호화에 필요한 암호화키를 관리하며, 사용이 완료된 데이터를 파악하고 관리하는 등의 다양한 방법들이 사용되고 있다. 이를 위하여 문서보안(DRM : Digital Rights Management), 데이터 손실 방지 솔루션(DLP : Data Loss Prevention), 국가정보원 검증필 암호 모듈 등의 다양한 보안 제품들이 출시되고 있다. 본 연구에서는 앞서 거론된 보안 모듈들과는 다르게 User Application이 OS Kernel을 통해 디스크의 파일에 접근할 때, 디스크의 특정 영역에 접근 권한을 부여하여 해당 User Application이 접근 권한이 있는지를 파악하고, 데이터 활용에 제약을 두는 방법과 특정 시스템 콜에 대한 접근을 제한하도록 하여 데이터를 보호하는 방법을 제시한다. 본 연구에서는 User Application이 내부적으로 기존과 동일하게 File을 엑세스하는 시스템 콜을 호출하지만, 호출된 시스템 콜을 가로채서 데이터 암호화/복호화 및 접근 통제를 추가로 처리한 후, 요청받은 파일 읽기/쓰기 업무를 수행하도록 하는 일종의 Secure OS Module 제시한다.

연구방법 및 선행연구 고찰

1. 연구의 범위와 방법

보안 데이터를 실제 사용하거나, 해당 데이터를 가공하는 프로그램을 개발하는 사람들 중 많은 사람들이 보안의 필요성에 대한 인식이 낮고, 보안 조치는 불편하고 어려운 것으로 이해하는 경향이 있다. 이에 따라 사용자의 사용성은 그대로 유지한 상태에서 데이터의 보안성을 강화할 방법을 제공하기 위하여, 시스템 콜을 가로채기해서 보안성 강화 처리 절차를 수행한 후에 기존 시스템 콜을 호출하는 방식으로 연구를 진행하였다. 본 연구는 리눅스 기반의 커널(2.6.x)을 기반으로 진행하였으며, 파일 접근과 관련된 시스템 콜만을 대상으로 한다. 표 1은 본 연구에서 통제 대상인 시스템 콜의 목록이다.

2. 선행연구 및 이론적 고찰

리눅스 보안 모듈(LSM:Linux Security Module)에 대한 선행 연구가 진행되었다. 리눅스 커널의 변화를 최소한으로 하는 강제적 접근 통제 모듈의 설계 방식을 분석하여 본 연구에서 고려되어야 하는 부분에 대한 검토가 계속해서

TABLE 1. List of file access system calls

num	function name	설명
3	read()	read from file
4	write()	write to file
5	open()	open file or device
6	close()	close file or device
8	creat()	create file or device
9	link()	make new name of file
10	unlink()	delete file or linked-file
11	execv()	execute process
12	chdir()	change working directory
14	mknod()	create normal or special file
15	chmod()	change access authority of file
16	chown()	change owner of file
38	rename()	change name or location of file
39	mkdir()	create directory
40	rmdir()	remove directory
60	umask()	make file mask
61	chroot()	change root directory
83	symlink()	make symbolic link of file
84	lstat()	get statue of file
85	readlink()	get file name of symbolic link related file
92	truncate()	change length of file
93	ftruncate()	change length of file
94	fchmod()	change access authority of file
95	fchown()	change owner or group of file

진행되었다. 또한, 보안 모듈을 커널 속에 필요에 따라 적재/제거하는 방식인 동적 적재 커널 모듈(LKM:Loadable Kernel Module) 방식에 대한 구조를 파악하였다. 리눅스에서 제공하는 모듈이라는 개념을 기반으로, 실행 중인 커널에 동적으로 연결하거나 제거할 수 있는 오브젝트 기반의 모듈을 제공하는 서비스를 구현하는 것을 분석하였다.

Data 보안을 위한 SecureOS Module 제시

1. 시스템 콜 분석을 통한 SecureOS Module 모델 구성

시스템 콜(System Call)은 응용프로그램에서 운영체제의 기능을 사용하기 위해서 호출하는 소프트웨어 인터럽트의 한 종류이다. 운영체제는 기본적으로 User Mode와 Kernel Mode로 분리되어 있으며, 보안 및 안전상의 문제로 User Mode에서는 프로세스 실행/종료나 I/O 작업이 불가하여 매우 제한적인 역할을 수행한다. 즉 User Mode만으로는 어떤 작업도 불가능하다고 보는 것이 맞다. 이에 User Mode는 Kernel Mode에 대신 업무를 요청하게 되는데, 이때 발생하는 소프트웨어 인터럽트의 한 종류가 시스템 콜이다.

시스템 콜은 OS와 응용 프로그램 간의 상호작용을 관리하고 운영 체제 서비스를 제공하는 중요한 메커니즘으로 파일시스템 접근을 관리하기 위해서도 시스템 콜은 핵심 역할을 수행한다. 표 2는 시스템 콜의 역할을 설명한 것이다.

소프트웨어 인터럽트에는 고유한 번호가 존재한다. 응용프로그램에서 호출된 시스템 콜은 커널 내부에 있는 IDT(Interrupt Descriptor Table)를 참조하여, 인터럽트 번호와 매핑된 서

TABLE 2. Roll of system call

roll	explanation
accessing and controlling of resources	Enables applications to access and control resources in the operating system. These resources include objects such as files and directories in the file system, and these resources can be accessed through system calls.
process management	Process management tasks – process creation, termination, suspension, resumption, and scheduling. This is the basis for several tasks, including file system access.
I/O managenet	Since file system access involves reading and writing data to disk, system calls manage and execute input and output operations in files and directories. This enables reading and writing of data, and creates and deletes files.
communication and networking	File systems can be used to copy or move files to other devices or systems, and system calls support these communications and networking operations. It is also used to transfer or share files to other systems.
security and privileges	Checking and managing access to the file system. When a user tries to read or modify a file, the system call can verify and reject this operation.
resource allocation and relase	The file system is used to allocate and release disk space. System calls provide the ability to resize files and directories and effectively utilize disk space.
error handling	To handle errors or exceptional situations that occur during file system access, system calls return error codes and provide an appropriate error handling mechanism.

비스 루틴을 실행한다. 본 연구는 이 시스템 콜를 가로채기하여 시스템 콜 테이블에 매핑된 서비스 루틴을 변경함으로써, 원래 호출된 함수가 아니라 다른 함수를 실행하도록 하는 것이다.

시스템 콜을 통해 파일시스템 접근을 제어하기 위한 권한 부여 및 권한 검사 메커니즘은 시스템의 보안 및 데이터 무결성을 유지하기 위해 핵심적인 역할을 수행한다. 이 메커니즘은 파일 및 디렉토리에 대한 접근 권한을 관리하고 파일 시스템에 대한 요청을 검사하여 허용되지 않은 접근을 방지할 수 있다. 일반적인 파일시스템은 읽기(Read)/쓰기(Write)/실행(Execute)과 같은 기본 권한을 소유자, 그룹 및 기타 사용자에게 설정할 수 있으며, 파일 또는 디렉토리에 대한 엑세스 요청을 받았을 때, 해당 요청을 호출한 사용자를 식별하여 권한을 확인한다. 이때 일반적인 권한 검사는 사용자 식별(요청을 보낸 사용자 식별), 파일 또는 디렉토리 권한 검사 후 권한 부여 여부(사용자가 요청한 읽기/쓰기/실행이 허용) 결정한다. chmod 및 chown 명령을 통해 파일시스템에서 파일의 권한을 수정하고 소유자를 변경할 수 있다. 그림 1은 SecureOS

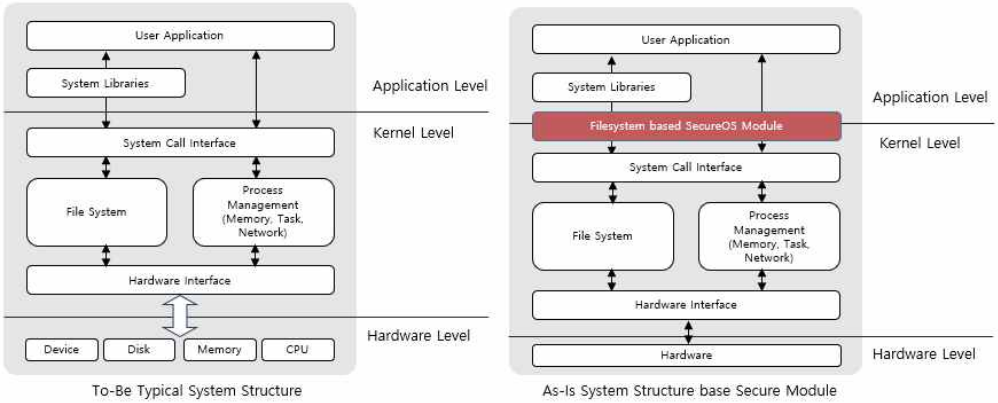

FIGURE 1. System architecture of a typical OS vs SecureOS Module

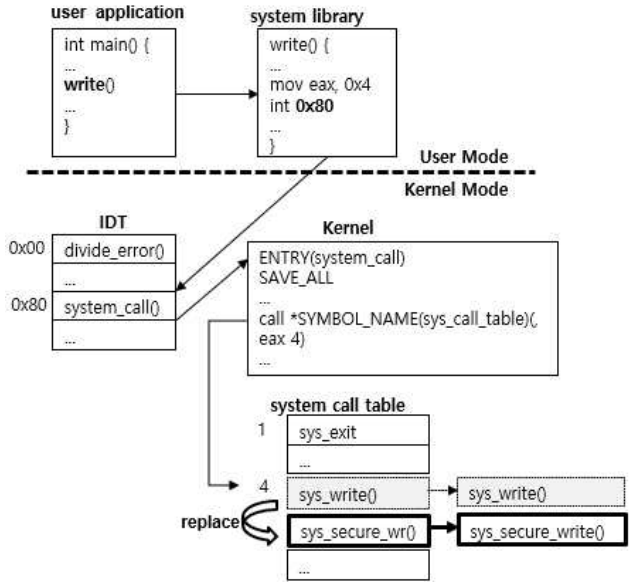

FIGURE 2. Flow of hooking system calls

Module이 적용된 형상을 보여준다.

이러한 권한 부여 및 권한 검사 메커니즘은 파일시스템의 기본 보안 구성 요소 중 하나이며, 파일 및 디렉토리에 대한 액세스를 효과적으로 제어함으로써 시스템 전체의 보안을 강화할 수 있다. 파일시스템에서의 권한 관리는 운영 체제와 응용 프로그램 간의 상호 작용을 통해 수행되며, 데이터 보안과 무결성을 지키기 위한 중요한 단계이다. 본 연구는 이와 같은 절차를 수행하는 시스템 콜을 가로채기하여 강화된 파일시스템의 보안성을 확보하는 것을 목표로 한다. 그림 2는 시스템 콜 중 write() 함수를 가로채기하는 과정을 표기한 것이다.

본 연구는 기존의 LKM(Loadable Kernel

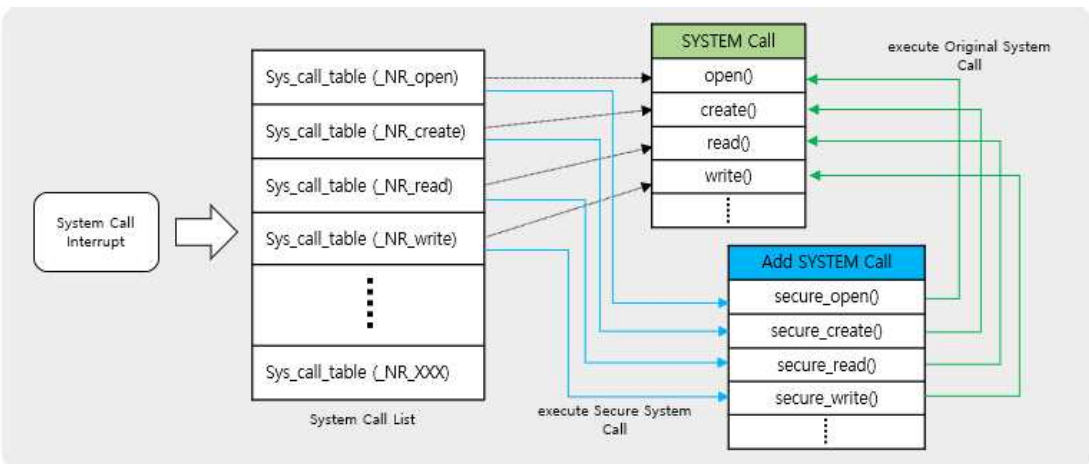

FIGURE 3. System Architecutre of SecureOS Module applied

Module)을 이용한 시스템 콜 가로채기 형태가 아닌, 사용자영역과 커널 영역 사이에서 통신을 가능하게 하는 시스템 콜을 가로채기함으로써 기존의 로직을 따르는 것이 아닌, 새롭게 구현된 보안 모듈을 실행하게 하여 파일시스템에 대한 보안 조치 후 반환하도록 한다. 이를 위하여 새롭게 구현된 보안 모듈이 포함된 시스템 콜을 추가 개발하고, 시스템 콜 매핑 리스트를 변경한다.

그림 3은 파일 암복호화 및 디렉토리 접근권한이 탑재된 보안 모듈이 포함된 SecureOS Module의 시스템 구조이다. 기존의 시스템 콜 매핑 리스트를 새롭게 추가된 함수가 호출되도록 변경하여 User Application은 동일한 함수를 호출하지만, 내부적으로는 변경된 함수가 호출되어, 소스 변경 없이 보안 모듈 기능을 제공할 수 있다.

보안 시스템 콜은 내부적으로 접근권한 통제 및 암복호화 처리 기능을 추가로 처리하며, 해당 작업이 완료된 후, 원래의 시스템 콜을 호출해서 요구된 기능을 제공하도록 한다.

2. SecureOS Module 제시

그림 4는 시스템 콜 가로채기를 통한 SecureOS Module 제시를 위한 시스템 구조도로 시스템 콜의 매핑 주소를 변경하여 User Application이 호출한 시스템 콜을 새롭게 구현된 보안 모듈을 통해 실행하도록 해서 파일시스템 접근 권한 통제 및 데이터 암복호화를 수행하는 SecureOS Module 방안을 제시한다.

1) 데이터 암호화 및 암호화키 관리

파일을 안전하게 저장하기 위하여 파일 입출력에 관련된 시스템 콜이 호출될 때, 해당 시스템 콜을 가로채기하여 파일을 암호화하는 방식을 제시한다. 파일 접근과 관련된 시스템 콜이 발생 시 SecureOS Module은 해당 시스템 콜이 요청한 대상 파일의 경로에 대한 암호키 관

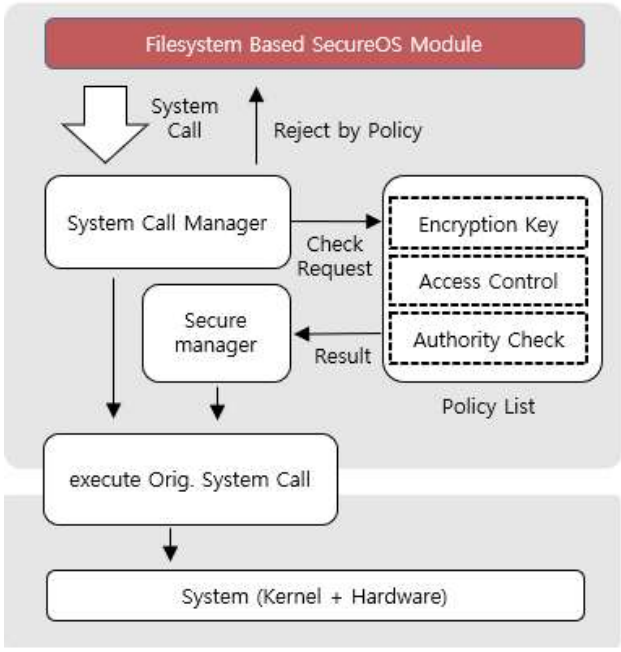

FIGURE 4. Procedure of system call hooking based SecureOS Module

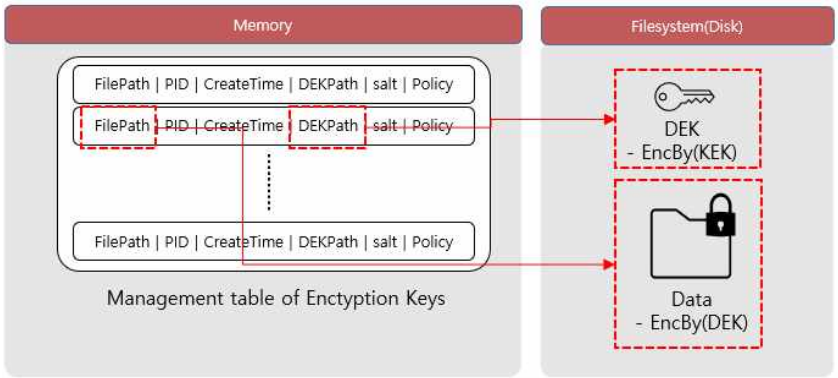

FIGURE 5. Data encryption and key management

리 테이블을 확인한다. 이때 해당 파일에 대한 정보가 없을 경우 새롭게 하나를 추가하고, 있을 경우 해당 데이터를 가지고 온다. 파일의 정보를 기반으로 PKCS#5를 통한 키 암호화 키(이하 KEK:Key Encryption Key)를 생성하고 32 bytes 이상의 Random 값으로 데이터 암호화 키(이하 DEK:Data Encryption Key)를 생성한다. 이후 KEK를 통해 DEK를 암호화하여 파일시스템에 저장하고 DEK 경로, KEK 유도값, PID, KEK 생성 시 사용된 salt 값을 암호키 관리 테이블에 저장한다. 이후 데이터의 Read/Write 시 해당 정보를 기반으로 데이터를 암복호화 한다. 그림 5는 Key 관리 Table을 도식화한 것이다.

KEK와 DEK를 사용하는 이중 키 암호화 방식은 데이터 보안을 강화하고, DEK의 안전한 보관과 관리를 가능하게 한다. 이 방식은 데이터 암호화와 관련된 키의 교환, 회전 및 관리를 단순화하여 데이터 보안을 효과적으로 관리할 수 있다. KEK에 의해 암호화된 DEK는 파일시스템에 암호화하여 저장되고, DEK를 암호화하는 암호키인 KEK는 별도로 저장하지 않는다. KEK를 유도값을 통해 생성하여 암호키에 대한 노출이 되지 않도록 최대한 고려했다. 또한 Policy 부분에 해당 파일의 생명 주기 등을 줄 수 있도록 확장할 수 있으며, 이에 따라 사용자가 파일에 접근하려고 할 때 해당 정책에 의해 암호키 관리 테이블의 정보 및 데이터를 삭제할 수 있어 보안 데이터의 휘발 처리가 가능하다.

2) 데이터 접근/실행 권한 관리

리눅스에서 파일의 종류는 디렉토리, 문자 디바이스 파일, 블록 디바이스 파일, 소켓, 심볼릭 링크로 구분되며, 각각에 대한 접근 권한을 설정할 수 있다. 파일의 접근 권한은 파일의 패스와 해당 파일에 설정된 사용자별 권한에 따라 읽기 허용, 쓰기 허용, 실행 허용, 파일 존재 여부 체크에 대한 권한 허용으로 세분화해서 설정할 수 있다. 사용자 접근 권한은 소유자, 해당 소유자가 속한 그룹, 기타 사용자로 구분할 수 있다. 각각의 사용자별로 파일 읽기/쓰기/실행에 대한 권한을 조합해서 설정할 수 있다.

표 3은 특수 권한으로 setuid, setgid, sticky bit을 설정해줄 수 있다. 프로세스가 생성되면

TABLE 3. Set user authority per file

Special Access Authority			Onwer Access Authority			Owner-group Access Authority			Other user Access Authority		
4	2	1	4	2	1	4	2	1	4	2	1
setuid	setgid	sticiky bit	r	w	x	r	w	x	r	w	x

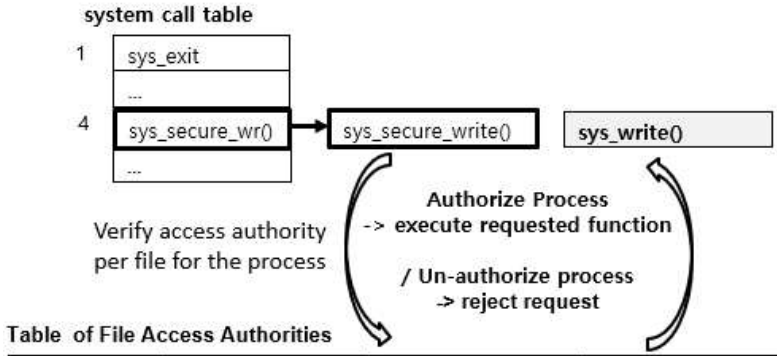

Table of File Access Authorities

file	Accessible User	Readable User	Writable User
/etc	root, admin, kim	root, admin	root
/sbin	root, admin, kim	root, admin	root
/sys	root, admin, kim	root, admin	root
/usr/lib64	root, admin, user1	root, admin, user1	root, admin
/usr/local	root, admin, user1	root, admin, user1	root, admin

FIGURE 6. Procedures for processing secure system calls

user id가 할당되며, 파일에 접근하는 시스템 콜을 호출한 프로세스의 id를 통해서, 해당 프로세스를 실행한 user id, group id 정보를 획득한 후, 파일에 설정된 소유자/그룹에 대한 권한과 비교한다. 또한 일반적으로 user id가 effective use id와 동일한 경우가 대부분이지만, dynamic protection system인 경우 다를 수 있기 때문에, uid가 euid와 다른 경우 euid도 확인 후 비교하는 절차를 수행해야 한다. Sticky bit가 설정된 디렉토리의 경우, 해당 디렉토리는 모든 user가 접근할 수 있지만, 자신이 생성하지 않은 파일을 삭제할 수는 없다.

응용 어플리케이션이 파일에 접근하기 위해서는 open/read/write/access와 같은 시스템 콜을 내부적으로 호출해야 하며, 위에서 거론된 사용자별 접근 권한을 시스템 콜 내부에서 비교 후 실제 요청한 기능을 수행하게 된다. 하지만 시스템 관리 명령어에 의해 파일 접근 권한은 외부에서 수정될 수 있으며, 그런 경우 실제로는 파일을 제어할 수 없는 사용자인 경우에도 파일을 제어할 수 있게 된다. 본 SecureOS Module 기반의 시스템 콜 가로채기 형태의 구현 제안에서는, 외부적으로 수정될 수 있는 파일의 접근 권한 설정과 별개로 내부적 파일별 접근 권한을 설정하고, 해당 권한을 확인하고 제어하는 추가 기능을 수행하는 시스템 콜을 기존 시스템 콜에 추가로 실행할 수 있게 한다. 그림 6은 본 연구에서 진행된 SecureOS Module 이 어떻게 권한 관리를 하는지 보여준다.

파일을 직접 엑세스하는 함수 외에도 파일이나 디렉토리의 정보 조회, 설정 변경, 생성, 삭제, 이동에 대한 모든 시스템 콜에 대해서 권한 확인 과정을 수행해야 한다. 다만 이미 권한확인 절차를 수행한 파일에 대한 정보 조회나, 재접근을 요청하는 경우에는 보안 처리 절차를 제외할 수 있다.

3) SecureOS Module Call Process

본 연구는 root의 권한을 가진 사용자라 할지라도 특정 파일시스템에 접근하거나 변경하는 것을 제한하기 위하여 시작되었다. 권한 제한을 두는 다양한 방법이 제시되었지만, 결국 사용자의 사용성에 의하여 해당 방안이 무효화되는 것을 확인하게 되었다. 이런 결과를 바탕으로 사

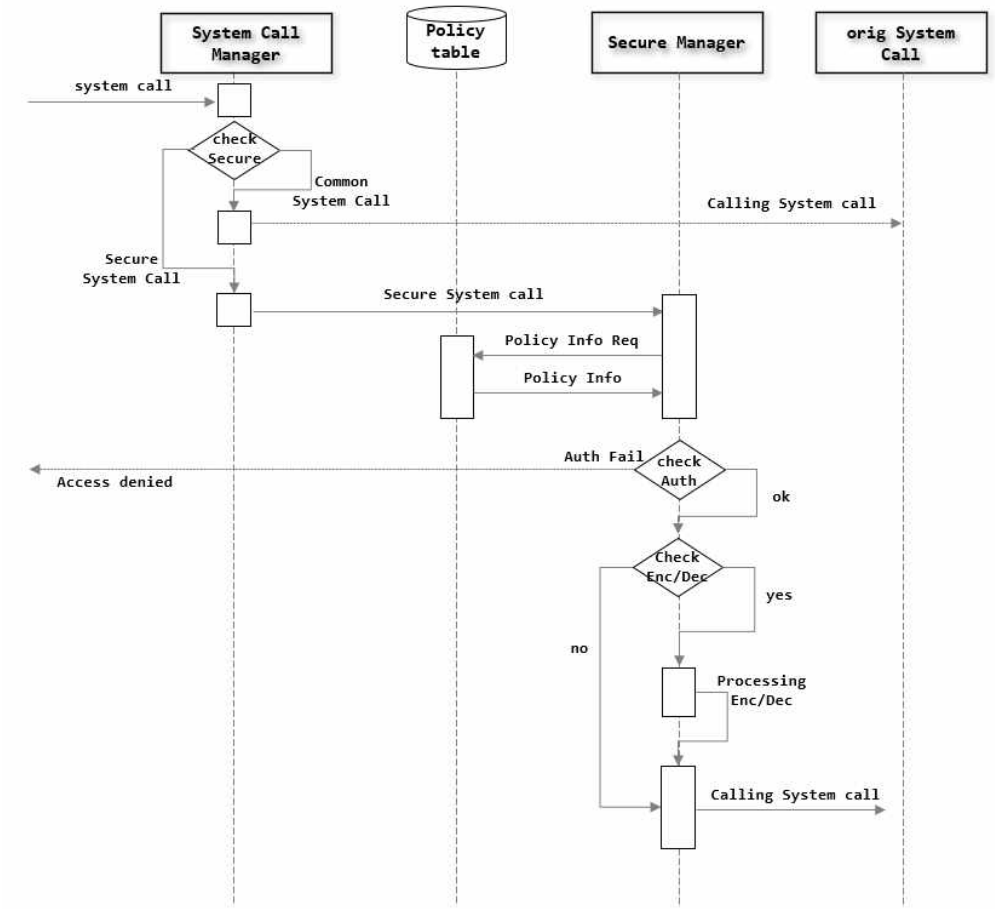

FIGURE 7. SecureOS Module Call Flow

용자의 사용성은 그대로 유지하면서 보안 기능을 제공하기 위해, 운영체제의 시스템 콜을 가로채서 일부를 변경하여 보안성을 추가 제공하는 방안을 고려하게 되었다. 그림 7은 본 연구에서 제시하고 있는 SecureOS Module의 Call Flow이다. 시스템 콜 가로채기를 할 수 있는 시스템 콜 매핑 테이블 수정 및 SecureOS Module을 커널 모듈로 설치한 후, 응용 프로그램이 운영체제에 서비스를 요청하기 위하여 시스템 콜을 호출하게 되면 System Call Manager는 시스템 콜 매핑 테이블에 의해 실제 실행되는 Funtion을 호출하게 된다. 이때 권한/보안 관리를 위해 선정된 특정 시스템 콜의 매핑 주소를 Secure Manager에서 제공하는 Function 주소로 변경해 놓으면, 응용 프로그램이 호출한 시스템 콜은 원래의 Function이 호출되는 것이 아니라 Secure Manager의 Function으로 호출된다. Secure Manager는 Policy Table에서 해당 Function의 권한을 체크하여 유효하지 않은 사용자 또는 권한이 허락되지 않은 디렉토리 접근 요청인지를 확인하여, 권한이 없을 경우 응용 프로그램에 접근 권한이 없음을 리턴한다. 정상적인 권한을 가지고 있다면 데이터 파일에 대한 접근인지를 확인한 후에 암복호화를 진행하여 기존의 시스템 콜을 호출한다.

4) SecureOS Module의 수행 결과 및 주요 성능 비교

3) 절에서 확인한 것과 같이 본 기술의 중요 포인트는 시스템 콜 매핑 테이블을 변경하여 원하는 시스템 콜을 재구성하는 것이다. 본 연구에서는 데이터 암복호화를 제외하면 성능 이슈는 무시할 수준으로 측정된다. 사용자와 디렉토리간 매핑 테이블을 통해 권한 체크 후 접근을 차단하거나, 기존의 시스템 콜을 호출하는 과정에 의한 지연은 거의 발생하지 않기 때문이다. 그러나 데이터를 읽고/쓰기를 할 때는 데이터의 암복호화를 수행하기 때문에 암복호화에 따른 지연이 발생할 수 있다. 이를 확인하기 위하여 표 4와 같이 아무 조치를 하지 않은 상태에서 write 속도, 별도의 암호모듈로 암호화한 후에 write 속도, SecureOS Module을 적용한 후에 write 속도를 측정했다. 암호화시에는 동일하게 SEED 128 알고리즘을 사용했고 암호화시 데이터 Chunk는 4096 bytes로 시험했다. 기타 시험 장비와 환경은 동일하게 설정했다.

표 4와 그림 8의 결과와 같이 별도의 암호모듈을 사용하여 암호화하고 저장하는 것과 SecureOS Module을 통해 암호화되어 저장하는 것의 성능 차이는 미미하다. 오히려 SecureOS Module을 사용했을 때, 초반 미세하게 낮은 수치를 보이기도 하는데, 이는 Encoder의 초기화하는 부분이 SecureOS Module을 로딩할 때 처리되면서 전체적인 성능의 차이를 두기 어려운 수준이 되었을 거라 판단된다. 그러나 용량이 큰 데이터를 인코딩할 경우 SecureOS Module의 시간이 미세하게 증가하는 것은 write 할 때마다 정책 비교에 대한 부하의 증가로 설명될 수 있다. 결론적으로 SecureOS Module 이 적용되었을 때, 의미 있는 성능의 저하가 발생하지 않음을 확인했다.

일반적인 어플리케이션 개발자 또는 다른 분야를 연구하는 연구진에게 데이터 보안을 어떻게 처리하는지 확인하였을 때, 일단 고려하지

TABLE 4. Speed Comparison by Storage Method

Data Size (MB)	Time (ms)		
	write	Enc(SEED)+write	SecureOS Module(SEED)
1	3	16	14
10	36	143	148
20	85	283	279
30	126	457	426
40	154	611	593
50	186	748	725
60	220	891	906
70	256	1061	1084
80	265	1228	1221
90	297	1407	1380
100	334	1535	1555
150	485	2303	2360
200	767	3059	3054
300	1081	4739	4622
400	1350	6174	6287
500	1495	7638	7771
600	1939	9050	9370
700	2508	10650	11105
800	2815	12013	12117
900	2988	13575	14093
1000	3293	15195	15734

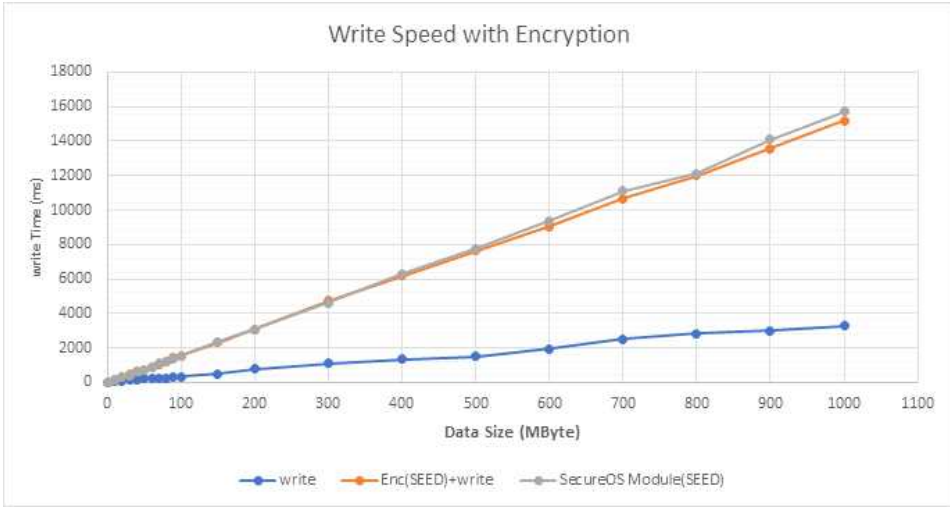

FIGURE 8. Write Speeds with Encryption

않는다와 데이터 암호화는 하지만 암호키 관리까지는 잘 고려하지 않는다는 것을 확인하였고, SecureOS Module을 제시했을 때, 사용성에서 기본적인 사용자가 특별히 고려하지 않아도 일정 수준의 보안성을 달성해주는 것에 대하여 만족감을 나타냈다. 그러나 커널을 설정해야 한다는 것에 부담감을 보이기도 했다.

결 론

파일시스템은 데이터 저장과 관리를 위한 핵심 구성 요소 중 하나이다. 그러나 파일시스템에 대한 무분별한 접근은 보안 문제와 데이터 손실의 위험을 초래할 수 있다. 이와 같은 문제를 해결하기 위하여 시스템 콜의 역할을 분석하여 보안성을 유지하면서 파일시스템의 접근을 효과적으로 제어하는 것을 목표로 하였다. 또한 보안 데이터를 사용하는 사용자 또는 해당 데이터를 가공하는 프로그램에서 데이터 보안을 유지하면서 개발하고 운영되는 것은 매우 어려운 일이다. 이런 이슈를 해결하기 위하여 사용자의 사용성은 최대한 유지한 상태에서 데이터 보안을 강화할 수 있는 방법을 고려하였다. 본 연구에서는 운영체제에서 발생하는 시스템 콜을 가로채고 수정할 수 있는 기술을 기반으로 보안 강화 목적의 SecureOS Module을 제시했다. 본 연구에서는 파일시스템 관련 시스템 콜에 대해 한정적으로 수행했지만, 모든 시스템 콜을 대상으로 한다면 이를 통해 다음과 같은 긍정적인 결과를 얻을 수 있을 것으로 기대된다.

첫째, 권한 제어 강화: 본 시스템을 활용하면 응용 프로그램이 파일시스템 또는 다른 운영 체제 리소스에 접근할 때 특정 권한을 부여하거나 거부할 수 있어 이는 보안 데이터 및 시스템 자원에 대한 접근을 미리 정의된 규칙에 따라 제어하고 강화할 수 있다.

둘째, 보안 정책 준수: 특정 보안 정책을 준수하도록 강제할 수 있다. 예를 들어, 파일 액세스 권한 또는 네트워크 트래픽 감사를 수행하고 이러한 정책에 대한 준수를 감시할 수 있다.

셋째, 실시간 모니터링 및 대응: 실시간으로 시스템 활동을 모니터링하고, 잠재적인 보안 위협을 식별할 수 있으며 이는 실시간으로 대응하여 보안 문제를 빠르게 해결할 수 있는 능력을 제공한다.

넷째, 외부 침입 탐지 및 방지: 외부 침입자가 시스템에 액세스하려고 시도하는 것을 탐지하고, 악의적인 시스템 콜을 사용을 방지하는데 사용될 수 있으며, 이는 시스템의 안정성을 유지하고 침입자의 활동을 중지하는 데 도움이

된다.

본 연구에서 제시한 SecureOS Module을 통해 권한 제어와 보안 강화를 달성할 수 있으며, 중요 데이터를 자동 암호화하고 안전하게 관리하여 데이터 보안을 더욱 강화할 수 있을 것으로 기대된다. 시스템 콜 가로채기를 통한 보안 관리 및 권한 제어는 보안 분야의 중요한 주제이며, 미래의 연구 및 개발에 대한 중요한 연구 주제로 계속해서 발전하고 있다. KAGIS

REFERENCES

Hong, S.S., Shin, H.J., Hwang, U.H., Chae, H.S., 2019, Study on Suitability for Web Service of River Geospatial Information, Journal of the Korean Association of Geographic Information Studies 22(2): 121-132 (홍성수, 신형진, 황의호, 채효석, 하천공간정보 웹 서비스의 적합성에 관한 연구, 한국지리정보학회, 22(2):121-132)

Hyun, G.N., Park, N.J., 2023, A Study on the Law Analysis and the Application Method of the Personal Information Protection Act for the Promotion of Data Opening in Public Institutions. Journal of the Korean Institute of Information Technology, 9-11 (현광남, 박남제 2023. A Study on the Law Analysis and the Application Method of the Personal Information Protection Act for the Promotion of Data Opening in Public Institutions. 한국정보기술학회 9-11).

Jeong, H.G., Kang, KT., 2022, Application monitoring system design and implementation using system call pattern, Journal of KIISE 32(10):795-801 (정해건, 강경태, 2022, 시스템 콜 호출 패턴을 이용한 애플리케이션 모니터링 시스템 설계 및 구현, 정보과학회 논문지, 49(10):795-801).

Jang, Y.G, Jeong, J.H., Lee, J.W., Kim, H.S., 2009. A Study on Optimal Technical Factors of USFSS Based on Integrated Technique of Wireless Communication and Location Awareness, Journal of the Korean Association of Geographic Information Studies 12(4):48-58 (장용구, 정재형, 이준우, 김현수. 2009. 무선통신 및 위치인식 통합기술을 활용한 지하구조물 현장지원시스템 최적 요소기술 연구, 한국지리정보학회 12(4):48-58).

Kim, B.S., Cho, J.S., 2018. Linux Kernel Attack of IoT Device using LKM-based System Call Hooking, Journal of the Korean Institute of Information Scientists and Engineers, 1195-1197 (김병선, 조진성 2018. Linux Kernel Attack of IoT Device using LKM-based System Call Hooking 한국정보과학회 1195-1197).

Kim, H.Y., Lee, S.H., 2008, A Study on the Application of Social Network Analysis for Expanding the use of Spatial Data in Local Government, Journal of the Korean Association of Geographic Information Studies 11(3):80-91 (김호용, 이성호. 2008. 지방자치단체의 공간 Data 활용 확대를 위한 Social Network Analysis의 적용방안 연구, 한국지리정보학회지 11(3):80-91).

Kim. J.Y., Kim, H.J., Yu, G.Y., 2022, A Study on Effective Real Estate Big Data Management Method Using Graph Database Model, Journal of the Korean Association of Geographic Information Studies 25(4): 163-180 (김주영, 김현정, 유기윤. 2022. 그래프 데이터베이스 모델을 이용한 효율적인 부동산 빅데이터 관리 방안에 관한 연구. 한국지리정보학회지 25(4):163-180).

Kim, S.S., Hong, C.S., 2014, Prevention of personal Information leakage through system

call hooking in Android Kernel, Journal of the Korean Institute of Information Scientists and Engineers, 1,015-1,017 (김성수, 홍충선. 2014. Prevention of personal Information leakage through system call hooking in Android Kernel, 한국정보과학회 학술발표논문집 1,015-1,017).

Lee, E.I., Kim, D.H., 2022, Serialization Method for large spatial data transmission of High Definition Map, Journal of the Korean Association of Geographic Information Studies 25(4):32-48 (이은일, 김덕호. 2022. 정밀도로지도의 대용량 공간데이터 교환을 위한 직렬화 기법 설계, 한국지리정보학회지. 25(4):32-48).

Lee, S.Y., Kim, H.G., Park, M.S.. 2022. Encryption key generation and encryption algorithm classification of encrypted data for smartphone app, Journal of the Korea Institute of Information Security & Cryptology, 32(6):17-22 (이신영, 김한결, 박명서 2022. 암호화된 스마트폰 앱 데이터에 대한 암호화 키 생성 및 암호 알고리즘 분류 정보보호학회지. 32(6):17-22).

USGS. 2015. Landsat 8(L8) Data Users Handbook. Department of the Interior US Geological Survey v1. https://www.usgs.gov/land-resources/nli/landsat/landsat-8-data-users-handbook. (August 15, 2019)

Zhang, Jun, SU, Purui, FENG, Dengguo, 2006, Design and implementation of intrusion detection system based on system-call, Journal of Computer Applications 26(9):2137-2139.