내부 커널 격리를 이용한 커널 보안 강화에 대한 연구 동향

구윤주 [1], 강하영 [2], 권동현 [3]
[1] 부산대학교 정보컴퓨터공학부 학부생
[2] 부산대학교 정보융합공학과 석사과정
[3] 부산대학교 정보컴퓨터공학부 교수(교신저자)

ventus07@pusan.ac.kr, rkdgkdud12345@pusan.ac.kr, kwondh@pusan.ac.kr

A Survey of Kernel Security Using Intra-Kernel Isolation

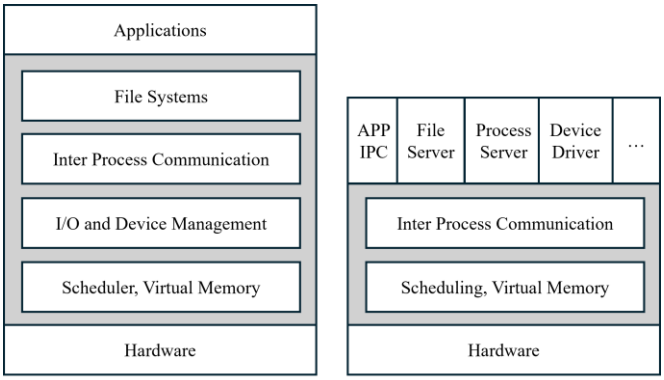
Yun-Ju Gu[1], Ha-Young Kang[2], Dong-Hyun Kwon[3]
[1]Dept. of Computer Science and Engineering, Pusan National University
[2]Dept. of Information Convergence Engineering, Pusan National University
[3]Dept. of Computer Science and Engineering, Pusan National University

요 약

커널은 운영 체제의 핵심 요소로 시스템 자원과 하드웨어를 관리하고 주요 시스템 서비스를 실행한다. 현대 프로세서의 대부분을 차지하는 모놀리식 커널은 모든 커널 기능이 동일한 메모리 공간과 동일한 높은 권한으로 실행되어 하나의 취약점이 시스템 전체를 위협할 수 있다. 이를 해결하기 위해 내부 커널 격리를 이용한 연구가 진행되어 왔다. 본 논문에서는 내부 커널 격리를 이용해 커널 보안을 강화한 최근 연구들에 대해 알아보았다.

1. 서론

커널은 운영 체제의 핵심 요소로서 시스템 자원과 하드웨어에 대한 접근을 제어하는 중요한 역할을 한다. 커널은 크게 모놀리식 커널과 마이크로 커널로 나뉜다[1]. 그림 1 에서 보여주듯이, 모놀리식 커널은 모든 주요 시스템 서비스를 커널 공간에서 실행하는 반면, 마이크로 커널은 프로세스 간 통신과 프로세스 관리 등 필수 기능만을 커널에서 처리하고, 나머지 서비스는 사용자 공간에서 실행하여 커널 크기를 줄인다. 현대 프로세서는 대부분 성능이 좋고 구현이 용이한 모놀리식 커널을 채택하여 사용한다. 그러나 모놀리식 커널은 모든 기능이 동일한 메모리 공간과 동일한 높은 권한 수준에서 실행되기 때문에, 하나의 취약점이 전체 시스템을 위험에 빠뜨릴 수 있다. 이러한 문제를 해결하기 위해 커널 내부의 각 구성 요소를 격리하는 방식으로 보안을 강화하여 특정 구성 요소의 오류나 취약점이 다른 부분에 영향을 미치지 않도록 차단하는 연구가 진행 중이다. 본 논문에서는 이 내부 커널 격리 기법을 통해 커널 보안을 강화한 최근 연구들을 분석하고 향후 연구방향을 제시한다.



(그림 1) 모놀리식 커널과 마이크로 커널

2. Framekernel

x86-64 에서 안전한 언어인 Rust 를 사용하여 내부 커널 권한 분리를 구현하고, 모놀리식 커널의 성능 효율성과 마이크로 커널의 보안성을 결합한 아키텍처이다[2]. Framekerne 은 커널을 특권 있는 OS 프레임워크와 비특권 OS 서비스로 나눈다. OS 프레임워크는 메모리 관리, CPU 상태 관리, 인터럽트 처리 등의 저수준 작업을 담당하며, 메모리 안전성과 제어 흐름 무결성을 보장하기 위해 검증된 API 와 함께 "unsafe"

Rust 코드를 사용한다. 반면 OS 서비스는 파일 시스템, 네트워크 스택, 디바이스 드라이버 등 다양한 OS 기능을 구현하며, 안전한 Rust 코드로 작성된다. OS 서비스가 저수준 작업을 수행하고자 할 때는 OS 프레임워크의 검증된 API 를 호출하여 필요한 작업을 수행하도록 한다. 이를 통해 동일한 메모리 공간에서 실행되더라도 Rust 의 언어 특성을 이용하여 두 커널을 성공적으로 분리하였고, 각 커널이 다른 권한을 가질 수 있도록 하였다.

3. GENESIS

내부 커널 격리를 이용한 다른 연구들과는 달리, 아키텍처에 종속되지 않는 권한 분리 설계를 목표로 한 연구이다[3]. GENESIS 는 모놀리식 커널을 내부 커널과 외부 커널로 나누어, 내부 커널에서는 보안이 중요한 기능을 수행하고 외부 커널에서는 일반적인 커널 작업을 수행하도록 하였다. 외부 커널에서는 메모리 관리 유닛(MMU)이나 페이지 테이블 등 중요한 시스템 자원에 접근할 수 없도록 하여 커널의 권한을 축소시켰다. 또한, 외부 커널과 내부 커널 간의 전환이 안전하게 이루어지도록 입/출구 게이트를 설계하여, 외부 커널이 보안이 중요한 작업을 직접 수행하는 것을 막았다. 이 과정에서 현대의 대부분의 상업 프로세서에서 제공하는 하드웨어 기능인 특권 접근 제한 기능(예: x86-64 의 SMAP, RISC-V 의 SUM)을 활성화하거나 비활성화하여 안전한 도메인 전환을 구현하였다. 이를 통해 외부 커널이 내부 커널의 시스템 자원에 직접 접근하는 것을 방지하고, 도메인 전환 중 외부 커널이 제어 흐름을 가로채는 것을 막아 도메인 전환의 원자성과 결정성을 보장하였다.

4. EC

커널과 사용자 코드가 같은 권한 레벨에서 실행되는 임베디드 시스템에서 메모리 보호를 강화하기 위해 내부 커널 격리 기술을 적용한 연구이다[4]. ARMv7-M 아키텍처에서 구현된 EC 는 펌웨어를 여러 개의 격리된 구성 요소로 분할하는 ECC 와 내부 커널 격리 기술을 이용해 런타임 메모리 보호를 구현한 마이크로 커널인 ECK 로 구성되어 있다. ECK 는 ARM Cortex-M 프로세서에서 제공하는 하드웨어 기능인 메모리 보호 장치(MPU)와 데이터 감시 및 추적 장치(DWT)를 활용하여 커널과 펌웨어가 동일한 권한 모드에서 실행되면서도 메모리 보호를 유지할 수 있도록 설계되었다. 먼저, MPU 를 사용하여 ECC 가 분할한 각 구성 요소의 메모리 영역에 접근 권한을 설정함으로써 각 구성 요소가 자신의 메모리 외부로 접근하지 못하도록 하였다. DWT 는 MPU 의 메모리 설정을 보호하기 위해 사용되며, 적법하지 않은 메모리 설정 변경 시도를 탐지하면 즉시 디버그 예외를 발생시킨다. 따라서 ECK 는 기존의 커널과 함께 동일한 권한 수준에서 실행되지만, 각 구성 요소를 분할하고 메모리 보호를 통해 성공적으로 각 요소 간의 격리를 보장하였다.

5. 결론

본 논문에서는 내부 커널 격리를 이용해 커널의 보호를 강화한 연구들에 대해 알아보았다. Framekernel 은 안전한 언어의 특성을 사용해 커널의 권한을 분리하였다. 하드웨어에 종속된 기능을 사용하여 구현한 것이 아니기 때문에 아키텍처에 구애받지 않고 적용 가능하지만 C/C++과같이 안전하지 않은 언어로 작성된 라이브러리와 함께 사용 시 충돌 가능성이 있다. GENESIS 의 경우 대부분의 하드웨어에 존재하는 기능을 사용하여 다양한 아키텍처와 호환되도록 구현하였으나 실사용을 위해서는 상당한 수정이 필요하다. EC 는 권한 분리 없이 내부 커널 격리를 구현하여 성능 오버헤드를 줄였으나 특정 하드웨어에서만 사용 가능하며 부족한 MPU 영역 개수 등 각 기능의 단점이 보안성에 영향을 미친다. 따라서 향후 연구에서는 이러한 단점을 보완하여 다양한 환경에서 실용성을 극대화하면서도 내부 커널 격리의 보안성을 더욱 강화하는 방향으로 나아갈 필요가 있다.

Acknowledgement

이 논문은 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2021-II210724, 임베디드 시스템 악성코드 탐지 · 복원을 위한 RISC-V 기반 보안 CPU 아키텍처 핵심기술 개발)

참고문헌

[1] ROCH, Benjamin. Monolithic kernel vs. Microkernel. *TU Wien*, 2004, 1.

[2] PENG, Yuke, et al. Framekernel: A Safe and Efficient Kernel Architecture via Rust-based Intra-kernel Privilege Separation. In: *Proceedings of the 15th ACM SIGOPS Asia-Pacific Workshop on Systems*. 2024. p. 31-37.

[3] LEE, Seongman, et al. GENESIS: A Generalizable, Efficient, and Secure Intra-kernel Privilege Separation. In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. 2024. p. 1366-1375.

[4] KHAN, Arslan; XU, Dongyan; TIAN, Dave Jing. Ec: Embedded systems compartmentalization via intra-kernel isolation. In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023. p. 2990-3007.