운영체제 및 애플리케이션의 보안 격리를 위한 LibOS 연구 조사

백재원[1], 조영필[2]
[1]한양대학교 정보보안학과 석사과정
[2]한양대학교 컴퓨터소프트웨어학과 교수
qor291@hanyang.ac.kr, ypcho@hanyang.ac.kr

Research and Anlysis of LibOS for Security Isolation in Operating System and Applications

Jae-Won Baek[1], KimYeong-Pil Cho[2]
[1]Dept. of Information Security, Hanyang University
[2]Dept. of Computer Software, Hanyang University

요 약

LibOS는 전통적인 운영체제의 커널을 응용 프로그램 수준의 라이브러리로 재구성한 시스템이다. 애플리케이션이 운영체제의 다양한 기능을 호출할 때 기존 운영체제처럼 무겁고 복잡한 커널과 직접 상호작용하는 대신, 가벼운 라이브러리 형태로 제공하여 시스템의 메모리 사용량을 줄이고 성능을 향상시킨다. 또한 각 애플리케이션이 독립적인 프로세스처럼 동작할 수 있도록 지원하여 보안 격리를 유지한다. 대표적으로 LibOS는 Intel SGX와 결합되어 시스템 호출을 줄이고, Enclave 내에서 대부분의 작업을 수행하여 공격 표면을 줄인다. 본 논문에서는 LibOS가 Intel SGX와 같은 보안 기술을 사용한 연구에 대해 소개한다.

1. 서론

LibOS(Library Operating System)는 전통적인 운영체제의 커널을 응용 프로그램 수준의 라이브러리로 재구성한 시스템이다. 이는 운영체제의 기능을 경량화하여, 각 애플리케이션이 실행될 때 독립적으로 필요로 하는 운영체제의 기능을 가볍게 제공하여, 기존 운영체제의 무거운 커널과 상호작용하는 대신 간결한 라이브러리 형태로 기능을 호출할 수 있게 해준다. LibOS의 특징 및 기능은 세 가지 측면으로 볼 수 있다.

보안 격리 강화. LibOS는 애플리케이션의 실행을 독립적인 프로세스처럼 동작할 수 있도록 격리되어 다른 프로세스와 상호작용을 최소화한다. 특히, Intel SGX(Secure Guard Extensions)[1]와 같은 하드웨어 보안 기술과 결합될 때, 신뢰할 수 없는 영역과 상호작용을 줄이고 Enclave 내에서 애플리케이션 외부 공격이나 악성 코드로부터 더 안전하게 실행할 수 있다.

효율성 및 성능 향상. LibOS는 파일 시스템, 네트워킹, 메모리 관리 등 일반적인 운영체제 기능을 응용 프로그램 수준에서 제공한다. 이를 통해 시스템의 메모리 사용량을 줄이고 성능을 향상시킬 수 있다.

멀티태스킹 및 자원관리. LibOS는 단일 프로세스뿐만 아니라 다중 프로세스 및 멀티태스킹을 효율적으로 지원한다. 이를 통해 애플리케이션이 동시에 여러 작업을 처리할 수 있으며 전통적인 운영체제와 비교했을 때 더 효율적인 자원 관리가 가능하다.

최근 LibOS의 발전은 주로 클라우드 컴퓨팅 및 기밀 컴퓨팅과 관련이 있다. 본 논문은 기밀 컴퓨팅 환경에서 민감한 데이터를 처리하는 Intel SGX 기반 LibOS의 연구에 대해 조사하였다.

2. Graphene

Graphene[2]은 실세계의 다중 프로세스 애플리케이션(셸, 웹 서버, 컴파일러 등)을 지원하는 리눅스 LibOS이다. Intel SGX와 같은 하드웨어 보안 기술과 결합하여 보안 격리를 구현하였다. RPC(Remote Procedure Call)을 사용하여 각 LibOS 인스턴스 간 통신을 담당하여 프로세스의 상호작용 간 보안 격리를 유지하면서 통신할 수 있게 한다. RPC은 네트워크를 통해 한 컴퓨터에서 다른 컴퓨터로 프로시저 호출을 요청하는 메커니즘으로 Graphene에서 각각의

LibOS 인스턴스가 협력하여 POSIX API와 같은 다중 프로세스 추상화를 구현하고 Unix 파이프와 유사한 바이트 스트림을 통해 서로 통신하고 다중 프로세스 간 상호작용을 관리한다.

PAL ABI (Platform Adaptation Layer)를 사용하여 운영체제와의 상호작용을 최소화하고, 호스트 운영체제에 독립적인 추상화를 제공한다. 또한 체크포인트 및 마이그레이션 기능을 통해 실행 중인 애플리케이션의 상태를 저장하고, 다른 환경으로 옮길 수 있어 클라우드 환경에서 효율적인 자원 관리를 지원한다. 이는 메모리 사용량을 줄이고, 애플리케이션의 성능을 극대화하는 역할을 한다. Graphene은 KVM과 같은 기존 가상화 기술과 비교했을 때, 메모리 사용량을 10배 이상 줄일 수 있다.

3. Graphene-SGX

Graphene-SGX[3]은 Graphene LibOS를 기반으로 개발된 수정되지 않은 리눅스 애플리케이션을 SGX에서 격리하는 프레임워크이다. 수정되지 않은 리눅스 바이너리를 실행하기 위해 동적 로딩과 런타임 링크를 지원한다. 각 애플리케이션에 대해 고유한 매니페스트를 사용하여 실행되는 바이너리와 동적 라이브러리의 무결성을 확인하고 SHA-256 해시를 사용해 신뢰할 수 있는 파일을 확인한다. Graphene-SGX는 단일 프로세스 애플리케이션이 시스템 호출을 Enclave 내부에서 처리하도록 설계되었다. 이러한 시스템 호출은 LibOS가 처리하며, PAL를 지원한다. 일부 파일 시스템 호출은 신뢰할 수 없는 OS의 응답 확인이 필요하며, 파일 읽기 작업은 OS에서 데이터를 가져와 해시를 통해 검증된다. 또한 fork 및 IPC 기능을 사용하여 새로운 Enclave를 생성하고 부모-자식 간 통신을 처리하는 다중 프로세스도 지원한다. 이는 로컬 이늦ㅇ과 TLS 연결을 사용하여 부모와 자식 Enclave 간에 안전한 통신이 보장된다.

4. Occlum

Occlum[4]은 기존 연구 Graphene-SGX와 같은 SGX LibOS에서 멀티 태스킹의 프로세스 생성과 통신 비용이 높고 파일 시스템 관리가 비효율적으로 인한 성능과 보안 문제를 해결한 논문이다. Occlum은 SFI(Software Fault Isolation) 기법을 활용한 SFI-Isolation Processe(SIP)을 구현하였다. 기존 연구 Graphene-SGX는 새로운 Enclave 생성, 로컬 인증, 상태 복제 등 복잡한 과정을 거쳐야 하지만, SIP는 이러한 과정 없이 생성되어 기존보다 높은 성능을 보인다. 프로세스 간 통신 측면에서는 기존의 암호화된 메시지 간의 통신에서 SIP 간 통신은 같은 주소 공간을 공유하므로 암호화 없이 간단한 데이터 복사가 가능하다. 또한 기존에는 데이터 동기화 문제로 인해 읽기 전용 파일 시스템만 지원하는 반면 SIP는 동일한 LibOS 인스턴스를 공유하므로 쓰기 가능한 암호화된 파일 시스템으로 안전하고 효율적인 멀티태스킹을 구현하였다.

Occlum은 MPX 기반 다중 도메인 SFI(MMDSFI)를 사용하여 SIP 격리를 강화하였다. MMDSFI는 프로세스 간의 메모리 및 제어 흐름을 격리하는 데 사용된다. 각 도메인마다 두 개의 메인 메모리 영역을 할당하며, 메모리 영역은 가드 영역으로 보호된다. 가드 영역은 SFI 기술에서 사용되는 기법으로 메모리 접근을 단순화하고 최적화를 제공한다.

5. Eloes

Eloes[5]는 Intel SGX 환경에서 Exit-Less 시스템 호출과 SUVM(Secure User-managed Virtual Memory)를 제공하는 프레임워크이다. Eleos는 RPC 메커니즘을 사용한다. Enclave 내부에서 시스템 호출이 발생하면 직접적으로 Enclave Exit가 발생하는 대신 해당 호출은 RPC 매커니즘을 통해 신뢰할 수 없는 영역에서 처리된다. 시스템 호출 요청은 외부의 워커 스레드로 전달되고, 그 스레드가 호출을 처리한 후 그 결과를 다시 Enclave 반환한다. 이 방식으로 SGX Enclave Ocall을 최소화하여 성능 저하를 방지한다.

Eleos는 사용자 레벨에서 관리되는 SUVM을 제공하여 페이지 폴트를 처리할 때 Enclave Exit를 하지 않고 소프트웨어적으로 처리하여, 메모리 관리가 가능하다. Spointer라고 불리는 활성 포인터를 통해 메모리 주소를 관리한다. Spointer는 일반 포인터처럼 사용되지만, 추가적으로 주소 변환 및 페이지 캐시 관리 기능을 제공한다. 이는 각 페이지의 주소를 로컬 변수에 캐시하여 자주 액세스하는 페이지에 대한 성능 최적화를 제공하여 성능을 높인다.

6. 결론

본 논문에서는 Intel SGX와 LibOS를 결합한 연구들을 조사하였다. 모든 논문은 Enclave 내에서 LibOS를 통해 각 애플리케이션을 독립적으로 실행

하여 다른 애플리케이션과 상호작용을 최소화하여 공격 표면을 줄이는 데 중점을 두고 있으며, Eleos의 Exit-Less 시스템 호출이나 Occlum의 SFI 격리 기술을 통해 성능 저하를 최소화하는 방법을 연구하였다. 또한 멀티 프로세스 및 다중 애플리케이션 환경에서 자원 격리 및 관리 문제를 해결하였고 이를 통해 복잡한 클라우드 환경이나 다중 Enclave 시나리오에서도 보안성을 유지하는 데 중점을 두고 있다.

현재 데이터 보호를 중점으로 많은 연구가 진행되고 있다. Intel SGX와 같은 TEE(Trusted Execution Environment) 환경에서 성능 최적화와 자원 관리, 보안 안정성을 효율적으로 해결하는 다양한 연구가 진행되고 있으며, 앞으로도 주된 관심을 가지고 지속적인 연구가 필요하다고 생각된다.

연구	특징	보안 메커니즘
Graphene	경량화된 LibOS	POSIX 및 RPC을 통한 시스템 호출의 보안성 강화
Graphene-SGX	동적 로딩, fork, IPC 지원	동적 라이브러리 로딩 보안성 강화
Occlum	SFI 기반의 프로세스 격리	SIP 및 MMDSFI 기반 보안
Eleos	Exit-Less 시스템 호출, SUVM 제공	RPC를 통한 시스템 호출 및 SUVM을 통한 페이지 보안 강화

<표 1> LibOS 연구의 특징 및 보안 메커니즘

이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구 결과 임 (No. RS-2024-00438729, 익명화된 기밀실행을 이용한 전주기적 데이터 프라이버시 보호 기술 개발)

참고문헌

[1] Intel Software Guard Extensions Programming Reference, Oct. 2014. Reference no. 329298-002US.

[2] Tsai, Chia-Che, et al. "Cooperation and security isolation of library OSes for multi-process applications." Proceedings of the Ninth European Conference on Computer Systems. 2014.

[3] Tsai, Chia-Che, Donald E. Porter, and Mona Vij. "{Graphene-SGX}: A practical library {OS} for unmodified applications on {SGX}." 2017 USENIX Annual Technical Conference (USENIX ATC 17). 2017.

[4] Shen, Youren, et al. "Occlum: Secure and efficient multitasking inside a single enclave of intel sgx." Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. 2020.

[5] Orenbach, Meni, et al. "Eleos: ExitLess OS services for SGX enclaves." Proceedings of the Twelfth European Conference on Computer Systems. 2017.