(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0032413
(43) 공개일자 2020년03월26일

(51) 국제특허분류(Int. Cl.)
G06F 21/53 (2013.01) *G06F 21/60* (2013.01)
(52) CPC특허분류
G06F 21/53 (2013.01)
G06F 21/602 (2013.01)
(21) 출원번호 10-2018-0111416
(22) 출원일자 2018년09월18일
심사청구일자 2018년09월18일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
양단아
서울특별시 은평구 수색로20길 5-5, 402호 (수색동, 동진아트빌)
도인실
인천광역시 연수구 센트럴로 194, 201동 2705호 (더샵센트럴파크2단지아파트)
채기준
서울특별시 강남구 강남대로136길 34, 402호 (논현동,월드노블레스빌)
(74) 대리인
윤귀상

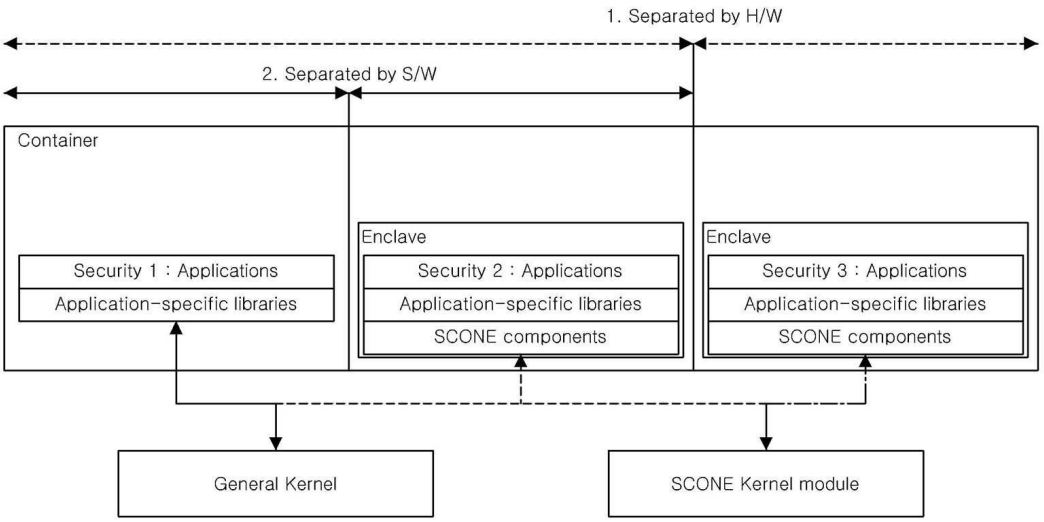
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 **리눅스 컨테이너 환경에서의 어플리케이션 처리 방법, 이를 수행하기 위한 기록매체 및 장치**

(57) 요 약

리눅스 컨테이너 환경에서의 어플리케이션 처리 방법, 이를 수행하기 위한 기록매체 및 장치가 개시된다. 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법은 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계 및 상기 어플리케이션에 부여된 보안 레벨에 따라 상기 어플리케이션의 코드를 그대로 실행하는 일반 커널 및 상기 어플리케이션의 코드를 암호화하여 실행하는 스콘(SCONE) 커널 중 적어도 하나의 커널을 선택하여 상기 어플리케이션을 실행하는 단계를 포함한다.

대 표 도 - 도3



(52) CPC특허분류
G06F 2221/2149 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 2016R1A2B4015899
부처명 과학기술정보통신부
연구관리전문기관 한국연구재단
연구사업명 기초연구사업(학술진흥)(2016)-중견연구자지원사업(총연구비1.5억초과~3억이하)
연구과제명 CPS 환경에서의 신뢰성있는 서비스 제공을 위한 보안기법 및 공격 대응 방안
기 여 율 1/1
주관기관 이화여자대학교 산학협력단
연구기간 2018.04.01 ~ 2019.02.28

명 세 서

청구범위

청구항 1

어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계; 및

상기 어플리케이션에 부여된 보안 레벨에 따라 상기 어플리케이션의 코드를 그대로 실행하는 일반 커널 및 상기 어플리케이션의 코드를 암호화하여 실행하는 스콘(SCONE) 커널 중 적어도 하나의 커널을 선택하여 상기 어플리케이션을 실행하는 단계를 포함하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법.

청구항 2

제1항에 있어서,

어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계는,

상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 3 단계로 부여하는 단계;

상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하지 않고, 상기 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 2 단계로 부여하는 단계; 및

상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건 및 소프트웨어 기반의 보안 요구 조건에 모두 해당하지 않는 경우, 상기 어플리케이션의 보안 레벨을 1 단계로 부여하는 단계를 포함하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법.

청구항 3

제2항에 있어서,

상기 어플리케이션에 부여된 보안 레벨에 따라 상기 어플리케이션의 코드를 그대로 실행하는 일반 커널 및 상기 어플리케이션의 코드를 암호화하여 실행하는 스콘(SCONE) 커널 중 적어도 하나의 커널을 선택하여 상기 어플리케이션을 실행하는 단계는,

상기 어플리케이션에 부여된 보안 레벨이 3 단계인 경우, 상기 스콘 커널을 선택하여 상기 어플리케이션을 실행하는 단계;

상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 일반 커널 및 상기 스콘 커널을 선택하여 상기 어플리케이션을 실행하는 단계; 및

상기 어플리케이션에 부여된 보안 레벨이 1 단계인 경우, 상기 일반 커널을 선택하여 상기 어플리케이션을 실행하는 단계를 포함하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법.

청구항 4

제3항에 있어서,

상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 일반 커널 및 상기 스콘 커널을 선택하여 상기 어플리케이션을 실행하는 단계는,

상기 어플리케이션의 함수(function) 중 하드웨어 기반의 보안 요구 조건이 설정된 함수를 상기 스콘 커널에서 실행하는 단계를 포함하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법.

청구항 5

제1항에 있어서,

상기 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계는,

상기 어플리케이션의 처리 전 과정이 Intel SGX(Software Guard Extensions)에 해당하는지 확인하여 상기 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건으로 분류하는 단계를 포함하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법.

청구항 6

제1항에 있어서,

어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계는,

상기 어플리케이션이 시큐어 코딩(Secure Coding)에 의해 개발되었는지 확인하여 상기 어플리케이션의 보안 요구 조건을 소프트웨어 기반의 보안 요구 조건으로 분류하는 단계를 포함하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법.

청구항 7

제1항 내지 제6항 중 어느 하나의 항에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법을 수행하기 위한, 컴퓨터 프로그램이 기록된 컴퓨터로 판독 가능한 기록 매체.

청구항 8

어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 어플리케이션 분류부;

상기 어플리케이션에 부여된 보안 레벨에 따라 상기 어플리케이션의 코드를 그대로 실행하는 일반 커널 및 상기 어플리케이션의 코드를 암호화하여 실행하는 스콘(SCONE) 커널 중 적어도 하나의 커널을 선택하는 커널 설정부; 및

선택한 커널에서 상기 어플리케이션을 실행하는 어플리케이션 실행부를 포함하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치.

청구항 9

제8항에 있어서,

상기 어플리케이션 분류부는,

상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 3 단계로 부여하고,

상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하지 않고, 상기 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 2 단계로 부여하며,

상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건 및 소프트웨어 기반의 보안 요구 조건에 모두 해당하지 않는 경우, 상기 어플리케이션의 보안 레벨을 1 단계로 부여하는 컨테이너 환경에서의 어플리케이션 처리 장치.

청구항 10

제9항에 있어서,

상기 커널 선택부는,

상기 어플리케이션에 부여된 보안 레벨이 3 단계인 경우, 상기 스콘 커널을 선택하고,

상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 일반 커널 및 상기 스콘 커널을 모두 선택하며,

상기 어플리케이션에 부여된 보안 레벨이 1 단계인 경우, 상기 일반 커널을 선택하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치.

청구항 11

제10항에 있어서,

상기 어플리케이션 실행부는,

상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 어플리케이션의 함수(function) 중 하드웨어 기반의 보안 요구 조건이 설정된 함수를 상기 스콘 커널에서 실행하고, 나머지 함수는 상기 일반 커널에서 실행하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치.

청구항 12

제9항에 있어서,

상기 어플리케이션 분류부는,

상기 어플리케이션의 처리 전 과정이 Intel SGX(Software Guard Extensions)에 해당하는지 확인하여 상기 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건으로 분류하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치.

청구항 13

제9항에 있어서,

상기 어플리케이션 분류부는,

상기 어플리케이션이 시큐어 코딩(Secure Coding)에 의해 개발되었는지 확인하여 상기 어플리케이션의 보안 요구 조건을 소프트웨어 기반의 보안 요구 조건으로 분류하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법, 이를 수행하기 위한 기록매체 및 장치에 관한 것으로서, 더욱 상세하게는 리눅스 컨테이너 환경에서 효율적인 리소스 관리를 위한 어플리케이션 처리 방법, 이를 수행하기 위한 기록매체 및 장치에 관한 것이다.

배 경 기 술

[0002] 클라우드 컴퓨팅은 한 사용자당 가상머신 하나를 생성하여, 어떤 호스트에서든지 사용자의 어플리케이션을 실행할 수 있도록 제공해주는 플랫폼이다.

[0003] 리눅스 컨테이너는 이러한 클라우드 컴퓨팅 환경에서 어플리케이션의 독립적인 실행을 도와줄 수 있으며, 일예로, Docker가 널리 사용되고 있다. 그러나, 리눅스 컨테이너는 독립적인 어플리케이션 실행을 위해 소프트웨어적인 보안 메커니즘에 의지하므로 보안성이 약하다는 문제점이 있다.

[0004] 이를 해결하기 위해 Intel SGX(Software Guard Extensions) 기반의 리눅스 컨테이너 환경이 제안된바 있다. Intel SGX는 모든 어플리케이션 코드를 암호화하여 실행하기 때문에 데이터의 기밀성과 무결성을 보장할 수 있다.

[0005] 그러나, 시계, 계산기, 캘린더 등의 위젯과 같이 보안이 불필요한 어플리케이션을 Intel SGX에서 실행하는 것은 메모리 비효율적이므로, 어플리케이션 실행의 보안성을 보장하면서, 보다 효율적인 리소스 사용이 가능한 어플리케이션 처리 방법이 제안될 필요가 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명의 일측면은 어플리케이션의 보안 요구 조건에 따라 분류되는 보안 레벨에 기반하여 일반 커널 및 어플리케이션의 실행 시 보안성을 보장하는 스콘 커널로 어플리케이션의 실행을 분담하는 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법, 이를 수행하기 위한 기록매체 및 장치를 제공한다.

과제의 해결 수단

[0007] 상기 과제를 해결하기 위한 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법은 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계 및 상기 어플리케이션에 부여된 보안 레벨에 따라 상기 어플리케이션의 코드를 그대로 실행하는 일반 커널 및 상기 어플리케이션의 코드를 암호화하여 실행하는 스콘(SCONE) 커널 중 적어도 하나의 커널을 선택하여 상기 어플리케이션을 실행하는 단계를 포함한다.

[0008] 한편, 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계는, 상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 3 단계로 부여하는 단계, 상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하지 않고, 상기 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 2 단계로 부여하는 단계 및 상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건 및 소프트웨어 기반의 보안 요구 조건에 모두 해당하지 않는 경우, 상기 어플리케이션의 보안 레벨을 1 단계로 부여하는 단계를 포함할 수 있다.

[0009] 또한, 상기 어플리케이션에 부여된 보안 레벨에 따라 상기 어플리케이션의 코드를 그대로 실행하는 일반 커널 및 상기 어플리케이션의 코드를 암호화하여 실행하는 스콘(SCONE) 커널 중 적어도 하나의 커널을 선택하여 상기 어플리케이션을 실행하는 단계는, 상기 어플리케이션에 부여된 보안 레벨이 3 단계인 경우, 상기 스콘 커널을 선택하여 상기 어플리케이션을 실행하는 단계, 상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 일반 커널 및 상기 스콘 커널을 선택하여 상기 어플리케이션을 실행하는 단계 및 상기 어플리케이션에 부여된 보안 레벨이 1 단계인 경우, 상기 일반 커널을 선택하여 상기 어플리케이션을 실행하는 단계를 포함할 수 있다.

[0010] 또한, 상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 일반 커널 및 상기 스콘 커널을 선택하여 상기 어플리케이션을 실행하는 단계는, 상기 어플리케이션의 함수(function) 중 하드웨어 기반의 보안 요구 조건이 설정된 함수를 상기 스콘 커널에서 실행하는 단계를 포함할 수 있다.

[0011] 또한, 상기 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계는, 상기 어플리케이션의 처리 전 과정이 Intel SGX(Software Guard Extensions)에 해당하는지 확인하여 상기 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건으로 분류하는 단계를 포함할 수 있다.

[0012] 또한, 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 단계는, 상기 어플리케이션이 시큐어 코딩(Secure Coding)에 의해 개발되었는지 확인하여 상기 어플리케이션의 보안 요구 조건을 소프트웨어 기반의 보안 요구 조건으로 분류하는 단계를 포함할 수 있다.

[0013] 또한 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법을 수행하기 위한, 컴퓨터 프로그램이 기록된 컴퓨터로 판독 가능한 기록 매체일 수 있다.

[0014] 한편, 본 발명의 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치는 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류하여 보안 레벨을 부여하는 어플리케이션 분류부, 상기 어플리케이션에 부여된 보안 레벨에 따라 상기 어플리케이션의 코드를 그대로 실행하는 일반 커널 및 상기 어플리케이션의 코드를 암호화하여 실행하는 스콘(SCONE) 커널 중 적어도 하나의 커널을 선택하는 커널 설정부 및 선택한 커널에서 상기 어플리케이션을 실행하는 어플리케이션 실행부를 포함한다.

[0015] 한편, 상기 어플리케이션 분류부는, 상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 3 단계로 부여하고, 상기 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하지 않고, 상기 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하는 경우, 상기 어플리케이션의 보안 레벨을 2 단계로 부여하며, 상기 어플리케이션의

보안 요구 조건이 하드웨어 기반의 보안 요구 조건 및 소프트웨어 기반의 보안 요구 조건에 모두 해당하지 않는 경우, 상기 어플리케이션의 보안 레벨을 1 단계로 부여할 수 있다.

[0016] 또한, 상기 커널 선택부는, 상기 어플리케이션에 부여된 보안 레벨이 3 단계인 경우, 상기 스콘 커널을 선택하고, 상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 일반 커널 및 상기 스콘 커널을 모두 선택하며, 상기 어플리케이션에 부여된 보안 레벨이 1 단계인 경우, 상기 일반 커널을 선택할 수 있다.

[0017] 또한, 상기 어플리케이션 실행부는, 상기 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 상기 어플리케이션의 함수(function) 중 하드웨어 기반의 보안 요구 조건이 설정된 함수를 상기 스콘 커널에서 실행하고, 나머지 함수는 상기 일반 커널에서 실행할 수 있다.

[0018] 또한, 상기 어플리케이션 분류부는, 상기 어플리케이션의 처리 전 과정이 Intel SGX(Software Guard Extensions)에 해당하는지 확인하여 상기 어플리케이션의 보안 요구 조건을 하드웨어 기반의 보안 요구 조건으로 분류할 수 있다.

[0019] 또한, 상기 어플리케이션 분류부는, 상기 어플리케이션이 시큐어 코딩(Secure Coding)에 의해 개발되었는지 확인하여 상기 어플리케이션의 보안 요구 조건을 소프트웨어 기반의 보안 요구 조건으로 분류할 수 있다.

발명의 효과

[0020] 본 발명에 따르면, 리눅스 컨테이너 환경에서 어플리케이션 실행의 기밀성, 무결성, 보안성을 보장할 뿐만 아니라 리소스 부담을 줄여 전체적인 시스템 효율을 향상시킬 수 있다.

도면의 간단한 설명

[0021] 도 1은 가상화 소프트웨어와 본 발명의 리눅스 컨테이너 환경을 비교하여 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시예에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치의 블록도이다.

도 3은 도 2에 도시된 어플리케이션 분류부에서 어플리케이션의 보안 요구 조건을 분류하는 것을 설명하기 위한 도면이다.

도 4는 도 2에 도시된 어플리케이션 분류부에서 어플리케이션의 보안 요구 조건을 분류하는 척도가 되는 Intel SGX(Software Guard Extensions)를 설명하기 위한 도면이다.

도 5는 본 발명의 리눅스 컨테이너 환경을 구축하는 운영체제에 마련되는 스콘 커널을 설명하기 위한 도면이다.

도 6은 본 발명의 일 실시예에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0022] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예와 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하거나 유사한 기능을 지칭한다.

[0023] 이하, 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.

[0024] 도 1은 가상화 소프트웨어와 본 발명의 리눅스 컨테이너 환경을 비교하여 설명하기 위한 도면이다.

[0025] 클라우딩 컴퓨팅은 운영체제가 설치되고, 하이퍼바이저나 가상 머신 모니터(VMM: Virtual Machine Monitor) 등 가상화를 담당하는 소프트웨어가 운영체제 위에 설치됨으로써 다양한 게스트의 운영체제 및 어플리케이션을 실행할 수 있도록 하는 방식이다.

[0026] 도 1의 (a)를 참조하면, 가상화 소프트웨어는 하나의 시스템에서 여러 사용자의 운영체제(Guest OS)가 동시에

실행될 수 있도록 제공한다.

[0027] 리눅스 컨테이너는 도 1의 (a)와 같은 가상화 환경에서 어플리케이션의 독립적인 실행을 도와주는 플랫폼으로, 도 1의 (b)를 참조하면, 운영체제(Host OS)의 커널을 공유하고, 시스템을 구성하는 다른 컴포넌트들로부터 어플리케이션 실행을 격리할 수 있다.

[0029] 도 2는 본 발명의 일 실시예에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치의 블록도이다.

[0030] 본 발명의 일 실시예에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치(이하, 어플리케이션 처리 장치)(1)는 도 1의 (b)와 같은 리눅스 컨테이너 환경에 적용되어 어플리케이션의 실행을 지원할 수 있다.

[0031] 이를 위해, 도 2를 참조하면, 본 발명의 일 실시예에 따른 어플리케이션 처리 장치(1)는 어플리케이션 분류부(10), 커널 설정부(30) 및 어플리케이션 실행부(50)를 포함할 수 있다.

[0032] 본 발명의 일 실시예에 따른 어플리케이션 처리 장치(1)는 별도의 단말이거나 또는 단말의 일부 모듈일 수 있다. 도 2에 도시된 어플리케이션 분류부(10), 커널 설정부(30) 및 어플리케이션 실행부(50)의 구성은 통합 모듈로 형성되거나, 하나 이상의 모듈로 이루어질 수 있다. 그러나, 이와 반대로 각 구성은 별도의 모듈로 이루어질 수도 있다.

[0033] 본 발명의 일 실시예에 따른 어플리케이션 처리 장치(1)는 이동성을 갖거나 고정될 수 있다. 본 발명의 일 실시예에 따른 어플리케이션 처리 장치(1)는 서버(server) 또는 엔진(engine) 형태일 수 있으며, 디바이스(device), 기구(apparatus), 단말(terminal), UE(user equipment), MS(mobile station), 무선기기(wireless device), 휴대기기(handheld device) 등 다른 용어로 불릴 수 있다.

[0034] 본 발명의 일 실시예에 따른 어플리케이션 처리 장치(1)는 어플리케이션 처리를 수행하기 위한 소프트웨어(어플리케이션)가 설치되어 실행될 수 있으며, 어플리케이션 분류부(10), 커널 설정부(30) 및 어플리케이션 실행부(50)의 구성은 어플리케이션 처리 장치(1)에서 실행되는 소프트웨어에 의해 제어될 수 있다.

[0035] 이하, 도 2에 도시된 본 발명의 일 실시예에 따른 어플리케이션 처리 장치(1)의 각 구성에 대하여 구체적으로 설명한다.

[0036] 어플리케이션 분류부(10)는 리눅스 컨테이너 환경에서 실행되는 어플리케이션의 보안 요구 조건을 확인하여 보안 레벨을 부여할 수 있다.

[0037] 어플리케이션의 보안 요구 조건은 크게 하드웨어 기반의 보안 요구 조건 또는 소프트웨어 기반의 보안 요구 조건으로 분류될 수 있다. 하드웨어 기반의 보안은 어플리케이션의 처리 전 과정을 암호화하는 방식일 수 있다. 소프트웨어 기반의 보안은 소프트웨어 개발의 코딩 과정에서 보안성을 확보하는 방식일 수 있다.

[0038] 본 발명에서는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하거나, 소프트웨어 기반의 보안 요구 조건에 해당하되 부분적으로 하드웨어 기반의 보안 요구 조건에 해당하거나, 보안 요구 조건이 없는 경우로 분류될 수 있다.

[0039] 하드웨어 기반의 보안을 요구하는 어플리케이션은 사용자의 공인인증서나 민감한 데이터를 다루는 어플리케이션으로, 예를 들면, 은행 어플리케이션 등에 해당할 수 있다.

[0040] 소프트웨어 기반의 보안을 요구하되 부분적으로 하드웨어 기반의 보안을 요구하는 어플리케이션은 관리자 계정으로 접속하는 페이지와 일반 사용자가 접속하는 페이지가 분리되는 어플리케이션으로, 예를 들면, 국내 공공기관의 공공 웹 사이트, C4I정보 체계, 해군함정전투체계, 지상사격통제체계 등에 해당할 수 있다.

[0041] 보안 요구 조건이 없는 어플리케이션은 예를 들면, 계산기, 캘린더 등의 위젯에 해당할 수 있다.

[0042] 보안 레벨은 보안 요구 조건을 나타낼 수 있는 레벨로 정의될 수 있다.

[0043] 본 발명에서는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는 경우, 보안 레벨이 가장 높은 3 단계로 설정될 수 있다.

[0044] 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하되 부분적으로 하드웨어 기반의 보안 요구 조건에 해당하는 경우, 보안 레벨이 2 단계로 설정될 수 있다.

[0045] 어플리케이션의 보안 요구 조건이 없는 경우, 보안 레벨이 가장 낮은 1 단계로 설정될 수 있다.

[0046] 어플리케이션 분류부(10)는 어플리케이션을 위와 같이 하드웨어 기반의 보안을 요구하는 어플리케이션, 소프트웨어 기반의 보안을 요구하되, 부분적으로 하드웨어 기반의 보안을 요구하는 어플리케이션 또는 보안을 요구하지 않는 어플리케이션으로 분류하고, 분류된 각 보안 요구 조건에 따라 1 단계 내지 3 단계의 보안 레벨을 부여할 수 있다. 이와 관련하여, 도 3 및 도 4를 참조하여 설명한다.

[0047] 도 3은 도 2에 도시된 어플리케이션 분류부에서 어플리케이션의 보안 요구 조건을 분류하는 것을 설명하기 위한 도면이고, 도 4는 도 2에 도시된 어플리케이션 분류부에서 어플리케이션의 보안 요구 조건을 분류하는 척도가 되는 Intel SGX(Software Guard Extensions)를 설명하기 위한 도면이다.

[0048] 도 3을 참조하면, 어플리케이션 분류부(10)는 먼저 리눅스 컨테이너 환경에서 실행할 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는지를 확인할 수 있다.

[0049] 이를 위해, 어플리케이션 분류부(10)는 어플리케이션의 처리 전 과정이 Intel SGX(Software Guard Extensions)에 해당하는지 확인할 수 있다.

[0050] Intel SGX는 EPM(Enclave Processor Mode)을 제공하여, 사용자가 신뢰된 하드웨어에 종속된 상태에서 어플리케이션이 실행되도록 하는 플랫폼이다. Intel SGX는 EPC(Enclave Page Cache)에서 어플리케이션의 모든 코드를 암호화하여 실행할 수 있다.

[0051] 도 4를 참조하면, Intel SGX는 하드웨어의 메모리에 EPC를 할당하고, 컨테이너의 Enclave에 있는 어플리케이션의 명령어들만 Enclave 메모리에 접근하여 실행할 수 있다. 이때, Enclave 메모리는 최고 권한을 가진 관리자도 접근이 불가능하다.

[0052] 소프트웨어 개발자들은 어플리케이션에서 Intel SGX에 의해 실행될 부분을 설정할 수 있으며, 이를 통해 어플리케이션의 코드 및 사용자 데이터를 보호할 수 있다.

[0053] 어플리케이션 분류부(10)는 어플리케이션의 처리 전 과정이 이러한 Intel SGX에 의해 실행되는 것으로 확인되는 경우, 해당 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는 것으로 분류하고, 보안 레벨 3 단계(Security 3)를 부여할 수 있다.

[0054] 또한, 어플리케이션 분류부(10)는 리눅스 컨테이너 환경에서 실행할 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하되, 부분적으로는 하드웨어 기반의 보안 요구 조건에 해당하는지를 확인할 수 있다.

[0055] 이를 위해, 도 3을 참조하면, 어플리케이션 분류부(10)는 어플리케이션의 처리 전 과정이 Intel SGX에 의해 실행되는 것으로 확인되지 않는 경우, 즉, 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건으로 분류되지 않는 경우, 해당 어플리케이션이 시큐어 코딩(Secure Coding)에 의해 개발되었는지를 확인하여 해당 어플리케이션을 소프트웨어 기반의 보안을 요구하되, 부분적으로는 하드웨어 기반의 보안을 요구하는 어플리케이션으로 분류할 수 있다.

[0056] 시큐어 코딩은 소프트웨어 품질이나 신뢰성 향상을 위하여 기존에 개발된 일반적인 코딩 규칙들을 기본적으로 참고하되, OWASP TOP 10과 같은 취약점 기준들을 참고하여 개발하는 방식이다.

[0057] 어플리케이션 분류부(10)는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건으로 분류되지 않고, 어플리케이션이 시큐어 코딩에 의해 개발된 것으로 확인되는 경우, 해당 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하되, 부분적으로는 하드웨어 기반의 보안 요구 조건에 해당하는 것으로 분류하고, 보안 레벨 2 단계(Security 2)를 부여할 수 있다.

[0058] 또한, 어플리케이션 분류부(10)는 리눅스 컨테이너 환경에서 실행할 어플리케이션이 보안 요구 조건이 없는 어플리케이션인지를 확인할 수 있다.

[0059] 이를 위해, 도 3을 참조하면, 어플리케이션 분류부(10)는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건으로 분류되지 않고, 해당 어플리케이션이 시큐어 코딩에 의해서도 개발되지 않은 것으로 확인되는 경우, 해당 어플리케이션은 보안 요구 조건이 없는 것으로 분류하여, 보안 레벨 1 단계(Security 1)를 부여할 수 있다.

[0060] 다시 도 2를 참조하면, 커널 설정부(30)는 어플리케이션에 부여된 보안 레벨에 따라 어플리케이션을 실행할 커널을 설정할 수 있다.

[0061] 본 발명의 리눅스 컨테이너 환경을 구축하는 운영체제(Host OS)는 일반 커널 이외에 스콘(SCONE) 커널이 마련될 수 있다. 일반 커널은 어플리케이션 코드를 그대로 실행하는 커널이고, 스콘 커널은 어플리케이션의 코드를 암호화하여 실행하는 커널일 수 있다. 여기서, 스콘(SCONE)은 Secure Linux Containers with Intel SGX의 약자로, 스콘 커널은 Arnautov 등에 의해 제안된 커널이다(Arnautov, Sergei, et al. "SCONE: Secure Linux Containers with Intel SGX," OSDI. Vol. 16. 2016.).

[0062] 도 5는 본 발명의 리눅스 컨테이너 환경을 구축하는 운영체제에 마련되는 스콘 커널을 설명하기 위한 도면이다.

[0063] 도 5를 참조하면, 스콘 커널은 Enclave의 어플리케이션이 Intel SGX에 의해 실행될 수 있도록 하는 커널로, 컨테이너의 Enclave에 독자적으로 개발한 SCONE 컴포넌트들을 두고, 이러한 SCONE 컴포넌트들을 암호화하여 system call request 및 response를 수행할 수 있다.

[0064] Intel SGX 실행을 위해서는 하드웨어(CPU) 및 SGX SDK로 프로그래밍 된 소프트웨어 등이 필요하므로, 어플리케이션의 수정이 필요한데, 스콘 커널은 도 5에 도시된 플랫폼을 구축하여 어플리케이션의 수정 없이 어플리케이션이 Intel SGX 에서 실행될 수 있도록 한다.

[0065] 커널 설정부(30)는 어플리케이션에 부여된 보안 레벨에 따라 일반 커널 및 스콘 커널 중 적어도 하나의 커널을 선택할 수 있다.

[0066] 커널 설정부(30)는 어플리케이션에 부여된 보안 레벨이 3 단계인 경우, 스콘 커널을 선택할 수 있다.

[0067] 커널 설정부(30)는 어플리케이션에 부여된 보안 레벨이 2 단계인 경우, 일반 커널 및 스콘 커널을 모두 선택할 수 있다.

[0068] 커널 설정부(30)는 어플리케이션에 부여된 보안 레벨이 1 단계인 경우, 일반 커널을 선택할 수 있다.

[0069] 어플리케이션 실행부(50)는 커널 설정부(30)에 의해 설정되는 일반 커널 또는 스콘 커널에서 어플리케이션을 실행할 수 있다.

[0070] 어플리케이션 실행부(50)는 어플리케이션의 보안 레벨이 3 단계로, 스콘 커널이 설정된 경우, 스콘 커널에서 해당 어플리케이션을 실행할 수 있다.

[0071] 어플리케이션 실행부(50)는 어플리케이션의 보안 레벨이 2 단계로, 일반 커널 및 스콘 커널이 모두 설정된 경우, 일반 커널 및 스콘 커널에서 해당 어플리케이션을 동시에 실행할 수 있다.

[0072] 구체적으로는, 어플리케이션 실행부(50)는 어플리케이션의 함수(function) 중 하드웨어 기반의 보안 요구 조건이 설정된 함수는 스콘 커널에서 실행할 수 있으며, 나머지 함수는 일반 커널에서 실행할 수 있다. 즉, 어플리케이션에서 Intel SGX를 이용해 실행되는 것으로 설정된 부분은 스콘 커널에서 실행할 수 있다.

[0073] 어플리케이션 실행부(50)는 어플리케이션의 보안 레벨이 1 단계로, 일반 커널이 설정된 경우, 일반 커널에서 해당 어플리케이션을 실행할 수 있다.

[0074] 이와 같이, 본 발명의 일 실시예에 따른 어플리케이션 처리 장치(1)는 어플리케이션의 보안 요구 조건에 따라 분류되는 보안 레벨에 기반하여 일반 커널 및 어플리케이션의 실행 시 보안성을 보장하는 스콘 커널로 어플리케이션의 실행을 분담할 수 있다. 이에 따라, 어플리케이션 실행의 보안성을 보장할 뿐만 아니라 리소스 부담을 줄여 전체적인 시스템 효율을 향상시킬 수 있다.

[0076] 도 6은 본 발명의 일 실시예에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법의 순서도이다.

[0077] 본 발명의 일 실시예에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법은 도 2에 도시된 어플리케이션 처리 장치(1)와 실질적으로 동일한 구성에서 진행될 수 있다. 따라서, 도 2의 어플리케이션 처리 장치(1)와 동일한 구성요소는 동일한 도면부호를 부여하고, 반복되는 설명은 생략한다.

[0078] 도 2를 참조하면, 어플리케이션 분류부(10)는 리눅스 컨테이너에서 실행할 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는지 확인할 수 있다(S100).

[0079] 어플리케이션 분류부(10)는 어플리케이션의 처리 전 과정이 Intel SGX에 의해 실행되는지를 확인하는 방식으로, 해당 어플리케이션이 하드웨어 기반의 보안을 요구하는지를 확인할 수 있다.

[0080] 어플리케이션 분류부(10)는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하는 것으

로 확인되는 경우(S100), 보안 레벨 3 단계를 부여할 수 있다(S200).

[0081] 어플리케이션 분류부(10)는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하지 않는 것으로 확인되는 경우(S100), 어플리케이션의 보안 요구 조건이 소프트웨어 기반의 보안 요구 조건에 해당하는지 확인할 수 있다(S300).

[0082] 어플리케이션 분류부(10)는 어플리케이션이 시큐어 코딩에 의해 개발되었는지를 확인하는 방식으로, 해당 어플리케이션이 소프트웨어 기반의 보안을 요구하되, 부분적으로는 하드웨어 기반의 보안을 요구하는 어플리케이션으로 확인할 수 있다.

[0083] 어플리케이션 분류부(10)는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하지 않고(S100), 소프트웨어 기반의 보안 요구 조건에 해당하는 것으로 확인되는 경우(S300), 보안 레벨 2 단계를 부여할 수 있다(S400).

[0084] 어플리케이션 분류부(10)는 어플리케이션의 보안 요구 조건이 하드웨어 기반의 보안 요구 조건에 해당하지 않고(S100), 소프트웨어 기반의 보안 요구 조건에도 해당하지 않는 것으로 확인되는 경우(S300), 보안 레벨 1 단계를 부여할 수 있다(S500).

[0085] 커널 설정부(30)는 어플리케이션에 보안 레벨 3 단계가 부여된 경우(S200), 해당 어플리케이션을 실행할 커널로 스콘 커널을 선택할 수 있다(S600).

[0086] 커널 설정부(30)는 어플리케이션에 보안 레벨 2 단계가 부여된 경우(S400), 해당 어플리케이션을 실행할 커널로 스콘 커널 및 일반 커널을 선택할 수 있다(S700).

[0087] 커널 설정부(30)는 어플리케이션에 보안 레벨 1 단계가 부여된 경우(S500), 해당 어플리케이션을 실행할 커널로 일반 커널을 선택할 수 있다(S800).

[0088] 어플리케이션 실행부(50)는 선택한 커널에서 어플리케이션을 실행할 수 있다(900).

[0089] 어플리케이션 실행부(50)는 보안 레벨 3 단계가 부여된 어플리케이션을 스콘 커널에서 실행할 수 있다.

[0090] 어플리케이션 실행부(50)는 보안 레벨 2 단계가 부여된 어플리케이션을 일반 커널 및 스콘 커널에서 동시에 실행할 수 있다. 어플리케이션 실행부(50)는 어플리케이션의 함수(function) 중 하드웨어 기반의 보안 요구 조건이 설정된 함수는 스콘 커널에서 실행할 수 있으며, 나머지 함수는 일반 커널에서 실행할 수 있다. 즉, 어플리케이션에서 Intel SGX를 이용해 실행되는 것으로 설정된 부분은 스콘 커널에서 실행할 수 있다.

[0091] 어플리케이션 실행부(50)는 보안 레벨 1 단계가 부여된 어플리케이션을 일반 커널에서 실행할 수 있다.

[0092] 이에 따라, 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법은 어플리케이션의 보안 요구 조건에 따라 분류되는 보안 레벨에 기반하여 일반 커널 및 어플리케이션의 실행 시 보안성을 보장하는 스콘 커널로 어플리케이션의 실행을 분담할 수 있다. 이에 따라, 어플리케이션 실행의 보안성을 보장할 뿐만 아니라 리소스 부담을 줄여 전체적인 시스템 효율을 향상시킬 수 있다.

[0094] 이와 같은, 본 발명의 일 실시예에 따른 리눅스 컨테이너 환경에서의 어플리케이션 처리 방법은 어플리케이션으로 구현되거나 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.

[0095] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.

[0096] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD 와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.

[0097] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처

리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0098] 이상에서는 실시예들을 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

부호의 설명

[0099] 1: 리눅스 컨테이너 환경에서의 어플리케이션 처리 장치

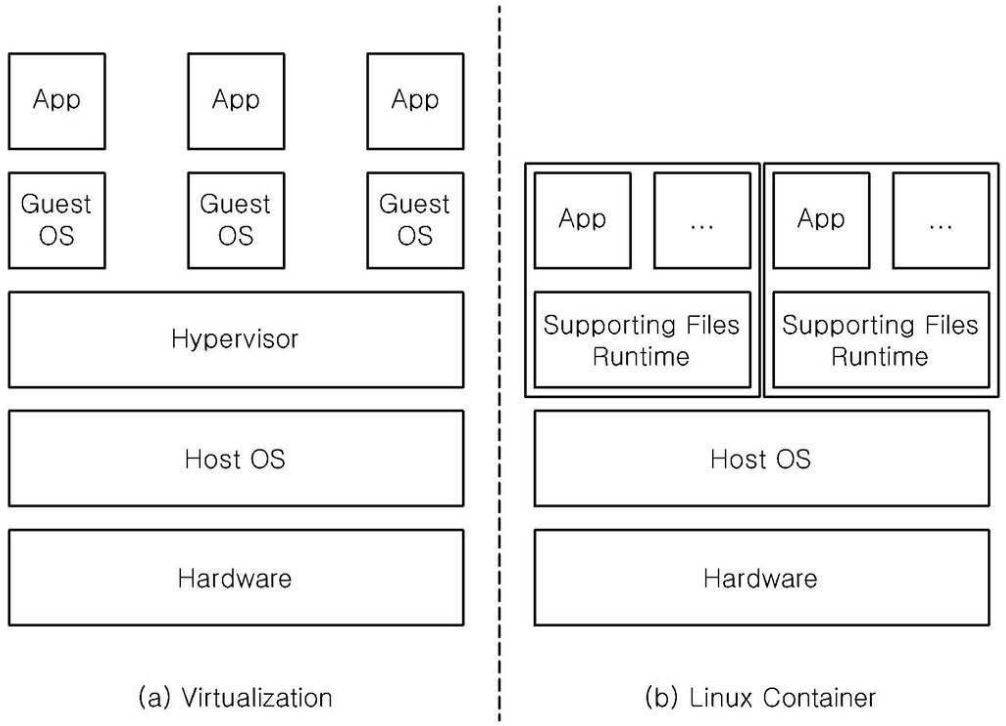
10: 어플리케이션 분류부

30: 커널 설정부

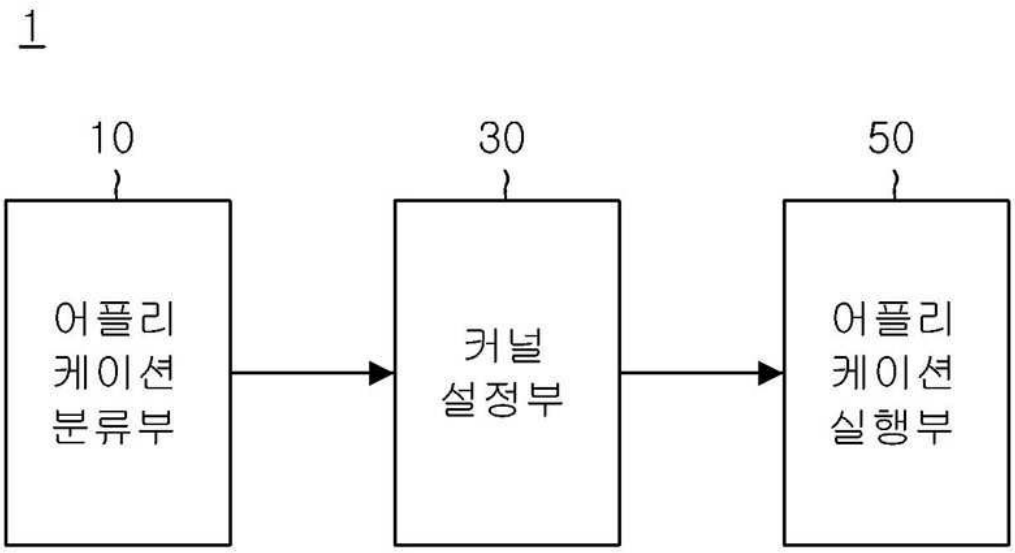
50: 어플리케이션 실행부

도면

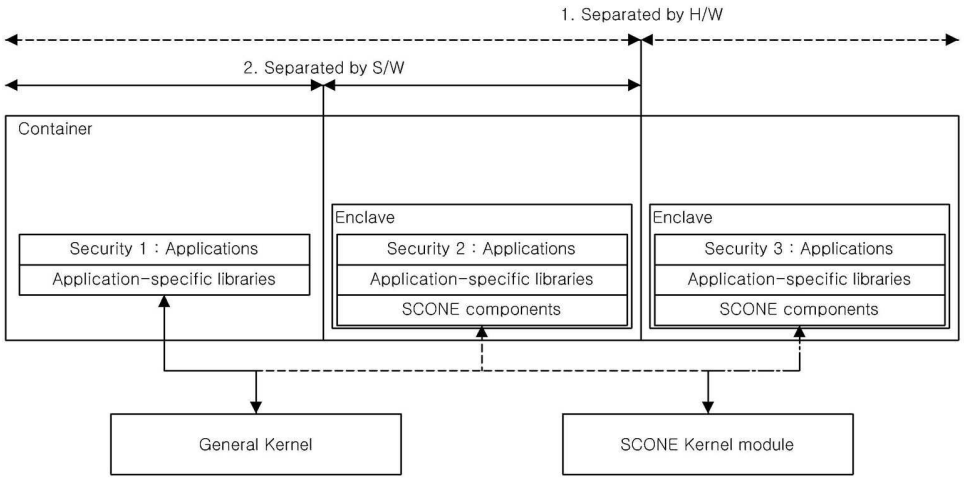
도면1



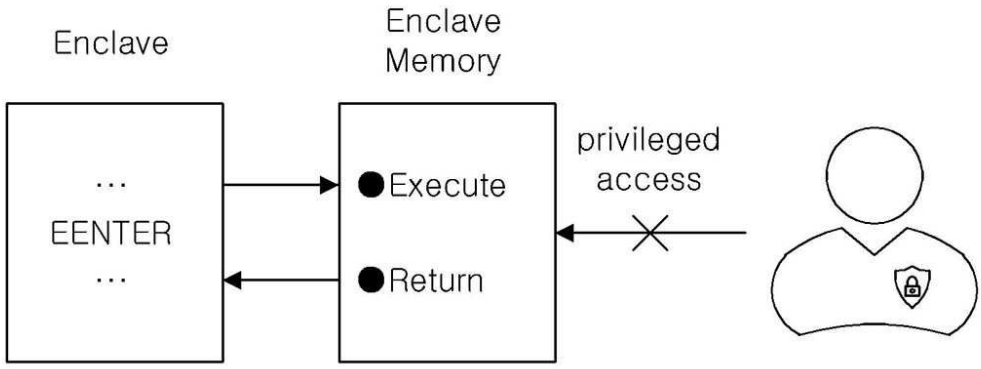
도면2


1
10
30
50
어플리
케이션
분류부
커널
설정부
어플리
케이션
실행부

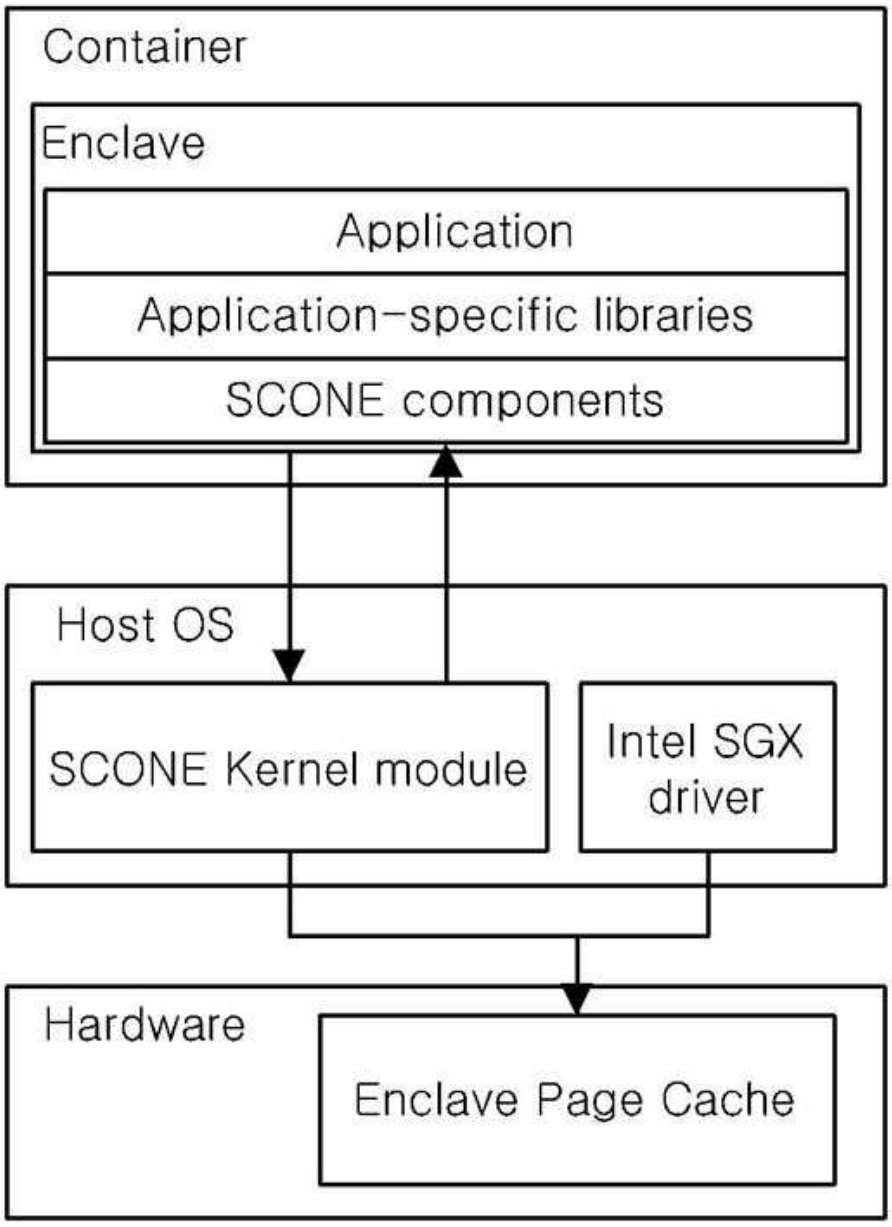
도면3


1. Separated by H/W
2. Separated by S/W
Container
Enclave
Enclave
Security 1 : Applications
Application-specific libraries
Security 2 : Applications
Application-specific libraries
SCONE components
Security 3 : Applications
Application-specific libraries
SCONE components
General Kernel
SCONE Kernel module

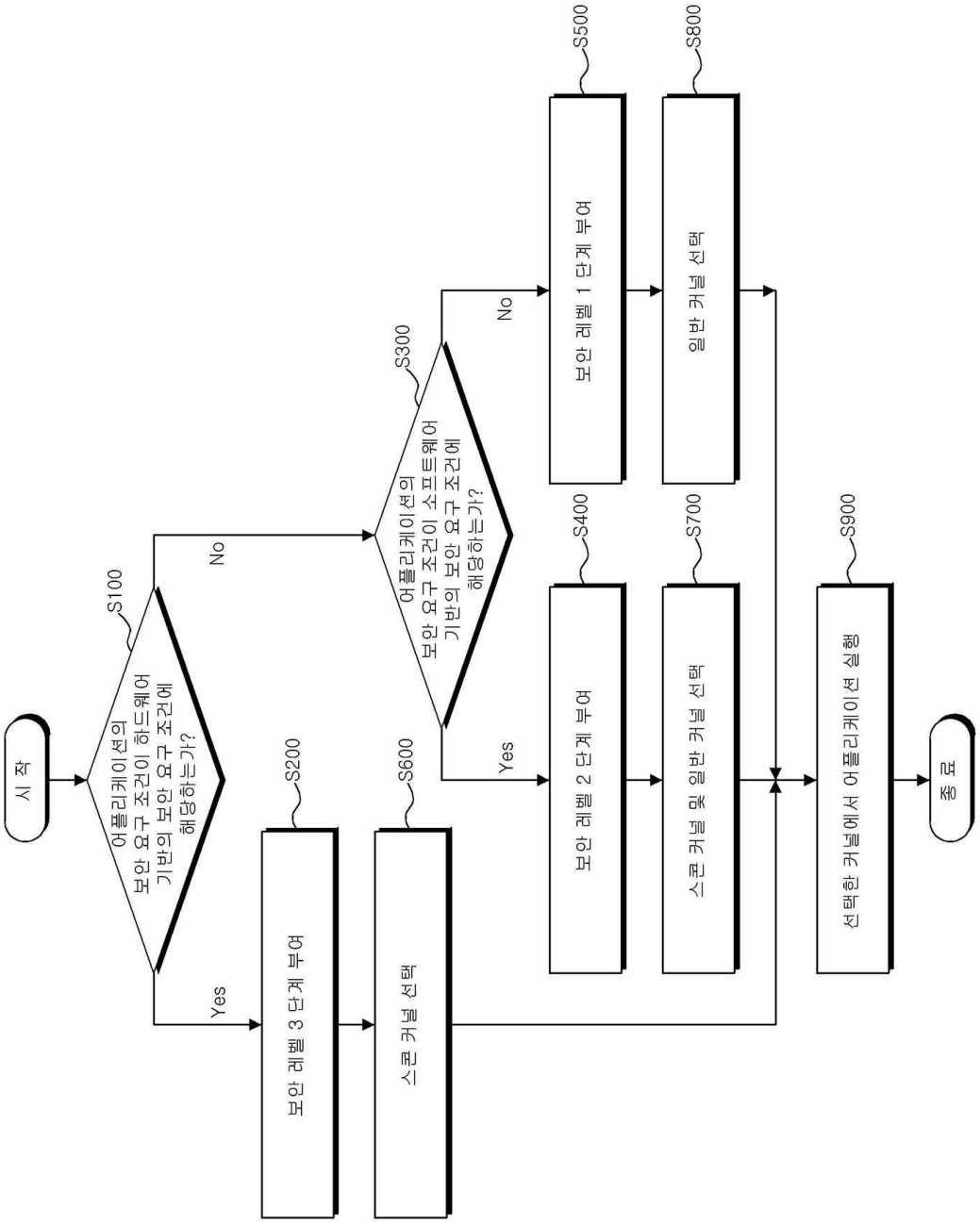
도면4


Enclave
Enclave Memory
...
EENTER
...
Execute
Return
privileged access

도면5


Container
Enclave
Application
Application-specific libraries
SCONE components
Host OS
SCONE Kernel module
Intel SGX driver
Hardware
Enclave Page Cache

도면6


시 작
S100
어플리케이션의
보안 요구 조건이 하드웨어
기반의 보안 요구 조건에
해당하는가?
Yes
No
S200
보안 레벨 3 단계 부여
S300
어플리케이션의
보안 요구 조건이 소프트웨어
기반의 보안 요구 조건에
해당하는가?
Yes
No
S400
보안 레벨 2 단계 부여
S500
보안 레벨 1 단계 부여
S600
스코 커널 선택
S700
스코 커널 및 일반 커널 선택
S800
일반 커널 선택
S900
선택한 커널에서 어플리케이션 실행
종 료