



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년03월25일
 (11) 등록번호 10-1961818
 (24) 등록일자 2019년03월19일

(51) 국제특허분류(Int. Cl.)
 G06F 12/14 (2006.01) G06F 12/02 (2018.01)
 (52) CPC특허분류
 G06F 12/1458 (2013.01)
 G06F 12/023 (2013.01)
 (21) 출원번호 10-2018-0136355
 (22) 출원일자 2018년11월08일
 심사청구일자 2018년11월08일
 (56) 선행기술조사문헌
 US20170364452 A1
 US20160092674 A1
 US10049211 B1
 KR101890125 B1

(73) 특허권자
 세종대학교산학협력단
 서울특별시 광진구 능동로 209 (군자동, 세종대학교)
 광운대학교 산학협력단
 서울특별시 노원구 광운로 20, 광운대학교 내 (월계동)
 (72) 발명자
 윤주범
 서울특별시 송파구 충민로4길 19, 704동 401호(장지동, 송파파인타운7단지)
 신영주
 경기도 의정부시 민락로 211, 115동 2101호(민락동, 호반베르디움1차)
 (74) 대리인
 두호특허법인

전체 청구항 수 : 총 10 항

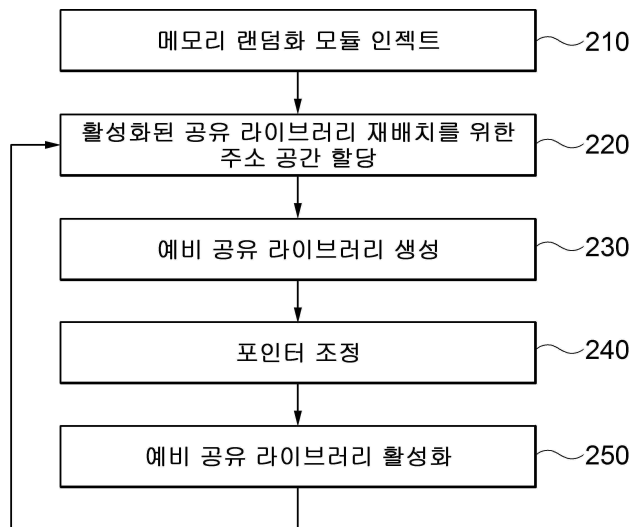
심사관 : 이상현

(54) 발명의 명칭 프로세스 중단 없는 메모리 랜덤화 방법 및 이를 수행하기 위한 컴퓨팅 장치

(57) 요약

프로세스 중단 없는 메모리 랜덤화 방법 및 이를 수행하기 위한 컴퓨팅 장치가 개시된다. 본 발명의 일 실시예에 따른 메모리 랜덤화 방법은, (a) 실행 중인 대상 프로그램에 메모리 랜덤화를 수행하기 위한 메모리 랜덤화 모듈을 인젝트(inject)하는 단계; (b) 상기 메모리 랜덤화 모듈이, 상기 대상 프로그램에 의해 이용되는 공유 라이브러리를 재배치하기 위한 임의의 주소 공간을 할당하는 단계; (c) 상기 메모리 랜덤화 모듈이, 상기 공유 라이브러리를 상기 할당된 메모리 주소 공간으로 복사하여 예비(stand-by) 공유 라이브러리를 생성하는 단계; (d) 상기 메모리 랜덤화 모듈이, 상기 공유 라이브러리에 대한 포인터를 수정하는 단계; 및 (e) 상기 메모리 랜덤화 모듈이, 상기 예비 공유 라이브러리를 활성화시키는 단계를 포함하되, 상기 (b) 단계 내지 상기 (e) 단계는 상기 대상 프로그램 실행 중 기 설정된 주기에 따라 반복 수행된다.

대표도 - 도2



- 이 발명을 지원한 국가연구개발사업
 과제고유번호 1711075702
 부처명 과학기술정보통신부
 연구관리전문기관 정보통신기술진흥센터
 연구사업명 정보통신기술인력양성(정보화)
 연구과제명 지능형 비행로봇 융합기술 연구
 기 여 율 6/10
 주관기관 세종대학교 산학협력단
 연구기간 2018.06.01 ~ 2021.12.31
- 이 발명을 지원한 국가연구개발사업
 과제고유번호 1711073404
 부처명 과학기술정보통신부
 연구관리전문기관 정보통신기술진흥센터
 연구사업명 SW전문인력역량강화
 연구과제명 SW중심대학(광운대)
 기 여 율 4/10
 주관기관 광운대학교 산학협력단
 연구기간 2018.01.01 ~ 2018.12.31
-

명세서

청구범위

청구항 1

- (a) 실행 중인 대상 프로그램에 메모리 랜덤화를 수행하기 위한 메모리 랜덤화 모듈을 인젝트(inject)하는 단계;
 - (b) 상기 메모리 랜덤화 모듈이, 상기 대상 프로그램에 의해 이용되는 공유 라이브러리를 재배치하기 위한 임의의 주소 공간을 할당하는 단계;
 - (c) 상기 메모리 랜덤화 모듈이, 상기 공유 라이브러리를 상기 할당된 메모리 주소 공간으로 복사하여 예비(stand-by) 공유 라이브러리를 생성하는 단계;
 - (d) 상기 메모리 랜덤화 모듈이, 상기 공유 라이브러리에 대한 포인터를 수정하는 단계; 및
 - (e) 상기 메모리 랜덤화 모듈이, 상기 예비 공유 라이브러리를 활성화시키는 단계를 포함하되,
- 상기 (b) 단계 내지 상기 (e) 단계는 상기 대상 프로그램 실행 중 기 설정된 주기에 따라 반복 수행되는 메모리 랜덤화 방법.

청구항 2

청구항 1에 있어서,

상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 실행 중 공유 라이브러리 형태로 메모리 공간 내에 인젝트되는 메모리 랜덤화 방법.

청구항 3

청구항 1에 있어서,

상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 프로세스 내에서 상기 메모리 랜덤화를 위해 생성된 스레드(thread)를 통해 상기 메모리 랜덤화를 수행하는 메모리 랜덤화 방법.

청구항 4

청구항 1에 있어서,

상기 (d) 단계는, 상기 공유 라이브러리의 각 함수에 대한 포인터가 상기 예비 공유 라이브러리의 함수 주소를 포인팅하도록 상기 공유 라이브러리의 각 함수에 대한 포인터를 수정하는 메모리 랜덤화 방법.

청구항 5

청구항 1에 있어서,

상기 (e) 단계는, PLT(procedure Linkage Table)에서 상기 공유 라이브러리의 함수들에 대한 주소를 예비 공유 라이브러리의 함수들에 대한 주소로 변경하는 메모리 랜덤화 방법.

청구항 6

실행 중인 대상 프로그램에 메모리 랜덤화 모듈을 인젝트(inject)하여 메모리 랜덤화를 수행하는 컴퓨팅 장치에 있어서, 상기 메모리 랜덤화 모듈은,

- (a) 상기 대상 프로그램에 의해 이용되는 공유 라이브러리를 재배치하기 위한 임의의 주소 공간을 할당하는 과정;
- (b) 상기 공유 라이브러리를 상기 할당된 메모리 주소 공간으로 복사하여 예비(stand-by) 공유 라이브러리를 생성하는 과정;
- (c) 상기 공유 라이브러리에 대한 포인터를 수정하는 과정; 및
- (d) 상기 예비 공유 라이브러리를 활성화시키는 과정을 상기 대상 프로그램 실행 중 기 설정된 주기에 따라 반복적으로 실행하기 위한 명령어들을 포함하는 컴퓨팅 장치.

청구항 7

청구항 6에 있어서,

상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 실행 중 공유 라이브러리 형태로 메모리 공간 내에 인젝트되는 컴퓨팅 장치.

청구항 8

청구항 6에 있어서,

상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 프로세스 내에서 상기 메모리 랜덤화를 위해 생성된 스레드(thread)를 통해 상기 메모리 랜덤화를 수행하는 컴퓨팅 장치.

청구항 9

청구항 6에 있어서,

상기 (c) 과정은, 상기 공유 라이브러리의 각 함수에 대한 포인터가 상기 예비 공유 라이브러리의 함수 주소를 포인팅하도록 상기 공유 라이브러리의 각 함수에 대한 포인터를 수정하는 컴퓨팅 장치.

청구항 10

청구항 6에 있어서,

상기 (d) 과정은, PLT(procedure Linkage Table)에서 상기 공유 라이브러리의 함수들에 대한 주소를 예비 공유 라이브러리의 함수들에 대한 주소로 변경하는 컴퓨팅 장치.

발명의 설명

기술 분야

[0001] 본 발명의 실시예들은 소프트웨어에 대한 메모리 공격을 방어하기 위한 기술과 관련된다.

배경 기술

[0002] 버퍼 오버플로우(buffer overflow) 공격, 포맷 스트링(format string) 공격 등과 같은 메모리 공격은 해커가 원격에서 관리자 권한의 셸(shell)을 획득하게 할 수 있을 정도로 심각한 위협이다. 이러한 메모리 공격을 막기 위해 ASLR(Address Space Layout Randomization)과 DEP(Data Execution Prevention)와 같은 방어 기법들이 개발되었다. 그러나, 코드 재사용(code reuse) 공격, ROP(Return-Oriented Programming) 공격 등과 같은 최신 메모리 공격 기술이 개발됨에 따라 이러한 방어 기법들 역시 우회 가능하게 되었다.

[0003] ROP 공격은 JIT-ROP, Blind-ROP 등의 공격으로 세분화되기도 한다. 이런 최신 공격 기술에 대응하기 위해 여러 방어 기법들이 제안되었으나, 이 방법들은 메모리 랜덤화를 위해 소프트웨어를 재컴파일하거나 프로세스를 중단시킨 후 다시 실행시켜야 하는 제한이 있어, 프로세스 중단 없이 수행되어야 하는 응용 프로그램(Mission Critical Application)에는 적합하지 않다는 문제점이 있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 미국 공개특허공보 US 2017/0364452 (2017.12.21. 공개)

발명의 내용

해결하려는 과제

[0005] 본 발명의 실시예들은 프로세스 중단 없이 메모리 랜덤화를 수행하기 위한 방법 및 이를 수행하기 위한 컴퓨터 장치를 제공하기 위한 것이다.

과제의 해결 수단

[0006] 본 발명의 일 실시예에 따른 메모리 랜덤화 방법은, (a) 실행 중인 대상 프로그램에 메모리 랜덤화를 수행하기 위한 메모리 랜덤화 모듈을 인젝트(inject)하는 단계; (b) 상기 메모리 랜덤화 모듈이, 상기 대상 프로그램에 의해 이용되는 공유 라이브러리를 재배치하기 위한 임의의 주소 공간을 할당하는 단계; (c) 상기 메모리 랜덤화 모듈이, 상기 공유 라이브러리를 상기 할당된 메모리 주소 공간으로 복사하여 예비(stand-by) 공유 라이브러리를 생성하는 단계; (d) 상기 메모리 랜덤화 모듈이, 상기 공유 라이브러리에 대한 포인터를 수정하는 단계; 및 (e) 상기 메모리 랜덤화 모듈이, 상기 예비 공유 라이브러리를 활성화시키는 단계를 포함하되, 상기 (b) 단계 내지 상기 (e) 단계는 상기 대상 프로그램 실행 중 기 설정된 주기에 따라 반복 수행된다.

[0007] 상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 실행 중 공유 라이브러리 형태로 메모리 공간 내에 인젝트될 수 있다.

[0008] 상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 프로세스 내에서 상기 메모리 랜덤화를 위해 생성된 스레드(thread)를 통해 상기 메모리 랜덤화를 수행할 수 있다.

[0009] 상기 (d) 단계는, 상기 공유 라이브러리의 각 함수에 대한 포인터가 상기 예비 공유 라이브러리의 함수 주소를 포인팅하도록 상기 공유 라이브러리의 각 함수에 대한 포인터를 수정할 수 있다.

[0010] 상기 (e) 단계는, PLT(procedure Linkage Table)에서 상기 공유 라이브러리의 함수들에 대한 주소를 예비 공유 라이브러리의 함수들에 대한 주소로 변경할 수 있다.

[0011] 본 발명의 일 실시예에 따른 컴퓨팅 장치는, 실행 중인 대상 프로그램에 메모리 랜덤화 모듈을 인젝트(inject)하여 메모리 랜덤화를 수행하며, 상기 메모리 랜덤화 모듈은, (a) 상기 대상 프로그램에 의해 이용되는 공유 라이브러리를 재배치하기 위한 임의의 주소 공간을 할당하는 과정; (b) 상기 공유 라이브러리를 상기 할당된 메모리 주소 공간으로 복사하여 예비(stand-by) 공유 라이브러리를 생성하는 과정; (c) 상기 공유 라이브러리에 대한 포인터를 수정하는 과정; 및 (d) 상기 예비 공유 라이브러리를 활성화시키는 과정 상기 대상 프로그램 실행 중 기 설정된 주기에 따라 반복적으로 실행하기 위한 명령어들을 포함한다.

[0012] 상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 실행 중 공유 라이브러리 형태로 메모리 공간 내에 인젝트될 수 있다.

[0013] 상기 메모리 랜덤화 모듈은, 상기 대상 프로그램의 프로세스 내에서 상기 메모리 랜덤화를 위해 생성된 스레드(thread)를 통해 상기 메모리 랜덤화를 수행할 수 있다.

[0014] 상기 (c) 과정은, 상기 공유 라이브러리의 각 함수에 대한 포인터가 상기 예비 공유 라이브러리의 함수 주소를 포인팅하도록 상기 공유 라이브러리의 각 함수에 대한 포인터를 수정할 수 있다.

[0015] 상기 (d) 과정은, PLT(procedure Linkage Table)에서 상기 공유 라이브러리의 함수들에 대한 주소를 예비 공유

라이브러리의 함수들에 대한 주소로 변경할 수 있다.

발명의 효과

[0016] 본 발명의 실시예들에 따르면, 대상 프로그램의 실행 중 인젝트되는 메모리 랜덤화 모듈을 이용하여 재컴파일이나 프로세스 중단 없이 메모리 랜덤화가 가능하도록 함으로써, 미션 크리티컬 애플리케이션과 같이 프로세스 중단 없이 수행되어야 하는 응용 프로그램에 대한 메모리 공격을 방어할 수 있게 된다.

도면의 간단한 설명

[0017] 도 1은 본 발명의 실시예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

도 2은 본 발명의 일 실시예에 따른 메모리 랜덤화 방법의 순서도

도 3은 본 발명의 일 실시예에 따른 메모리 랜덤화를 설명하기 위한 예시도

발명을 실시하기 위한 구체적인 내용

[0018] 이하, 도면을 참조하여 본 발명의 구체적인 실시형태를 설명하기로 한다. 이하의 상세한 설명은 본 명세서에서 기술된 방법, 장치 및/또는 시스템에 대한 포괄적인 이해를 돕기 위해 제공된다. 그러나 이는 예시에 불과하며 본 발명은 이에 제한되지 않는다.

[0019] 본 발명의 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 본 발명의 실시예들을 기술하기 위한 것이며, 결코 제한적이어서는 안 된다. 명확하게 달리 사용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 특성들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 특성, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0020] 도 1은 본 발명의 실시예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도이다. 도시된 실시예에서, 각 컴포넌트들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 이하에 기술되지 않은 것 이외에도 추가적인 컴포넌트를 포함할 수 있다.

[0021] 도시된 컴퓨팅 환경(10)은 컴퓨팅 장치(12)를 포함한다. 컴퓨팅 장치(12)는 적어도 하나의 프로세서(14), 컴퓨터 판독 가능 저장 매체(16) 및 통신 버스(18)를 포함한다. 프로세서(14)는 컴퓨팅 장치(12)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로세서(14)는 컴퓨터 판독 가능 저장 매체(16)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 상기 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨터 실행 가능 명령어는 프로세서(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0022] 컴퓨터 판독 가능 저장 매체(16)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능 저장 매체(16)에 저장된 프로그램(20)은 프로세서(14)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독 가능 저장 매체(16)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 컴퓨팅 장치(12)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0023] 통신 버스(18)는 프로세서(14), 컴퓨터 판독 가능 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0024] 컴퓨팅 장치(12)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(22) 및 하나 이상의 네트워크 통신 인터페이스(26)를 포함할 수 있다. 입출력 인터페이스(22) 및 네트워크 통신 인터페이스(26)는 통신 버스(18)에 연결된다. 입출력 장치(24)는 입출력 인터페이스(22)를 통해 컴퓨팅

장치(12)의 다른 컴포넌트들에 연결될 수 있다. 예시적인 입출력 장치(24)는 포인팅 장치(마우스 또는 트랙패드 등), 키보드, 터치 입력 장치(터치패드 또는 터치스크린 등), 음성 또는 소리 입력 장치, 다양한 종류의 센서 장치 및/또는 촬영 장치와 같은 입력 장치, 및/또는 디스플레이 장치, 프린터, 스피커 및/또는 네트워크 카드와 같은 출력 장치를 포함할 수 있다. 예시적인 입출력 장치(24)는 컴퓨팅 장치(12)를 구성하는 일 컴포넌트로서 컴퓨팅 장치(12)의 내부에 포함될 수도 있고, 컴퓨팅 장치(12)와는 구별되는 별개의 장치로 컴퓨팅 장치(12)와 연결될 수도 있다.

- [0025] 도 2은 본 발명의 일 실시예에 따른 메모리 랜덤화 방법의 순서도이다.
- [0026] 도 2에 도시된 방법은 예를 들어, 도 1에 도시된 컴퓨팅 장치(12)에 의해 수행될 수 있다.
- [0027] 도 2를 참조하면, 우선, 컴퓨팅 장치(12)는 실행 중인 대상 프로그램에 메모리 랜덤화 모듈을 인젝트(inject)한다(210).
- [0028] 구체적으로, 메모리 랜덤화 모듈은 공유 라이브러리(shared library) 형태로 대상 프로그램의 프로세스 실행 중 메모리 공간 내에 인젝트될 수 있다.
- [0029] 한편, 인젝트된 메모리 랜덤화 모듈은 대상 프로그램의 프로세스 내에서 메모리 랜덤화를 위한 스레드를 생성하여 메모리 랜덤화를 수행할 수 있다. 구체적으로, 메모리 랜덤화 모듈에 의해 수행되는 메모리 랜덤화는 대상 프로그램의 프로세스 내에서 메인 스레드(thread)와 구별되는 별도의 스레드를 통해 수행할 수 있다. 또한, 메모리 랜덤화 모듈이 대상 프로그램에 인젝트된 경우, 인젝트된 메모리 랜덤화 모듈의 데이터 구조, 클래스, 변수 등을 초기화하는 과정이 수반될 수 있다.
- [0030] 한편, 메모리 랜덤화 모듈이 인젝트된 경우, 메모리 랜덤화 모듈은 대상 프로그램에 의해 이용되고 있는 공유 라이브러리를 재배치하기 위한 임의의 주소 공간을 할당한다(220).
- [0031] 이후, 메모리 랜덤화 모듈은 공유 라이브러리를 할당된 주소 공간으로 복사하여 예비(stand-by) 공유 라이브러리를 생성한다(230).
- [0032] 이후, 메모리 랜덤화 모듈은 공유 라이브러리에 대한 포인터를 수정한다(240).
- [0033] 구체적으로, 메모리 랜덤화 모듈은 공유 라이브러리의 각 함수에 대한 포인터가 예비 공유 라이브러리의 함수 주소를 포인팅하도록 공유 라이브러리의 각 함수에 대한 포인터를 수정할 수 있다.
- [0034] 이후, 메모리 랜덤화 모듈은 예비 공유 라이브러리를 활성화시킨다(250).
- [0035] 구체적으로, 메모리 랜덤화 모듈은 대상 프로그램의 메인 스레드와 메모리 랜덤화를 위한 스레드를 동기화하고, PLT(Procedure Linkage Table)에서 공유 라이브러리의 함수들에 대한 주소를 예비 공유 라이브러리의 함수들에 대한 주소로 변경함으로써, 예비 공유 라이브러리를 활성화시킬 수 있다.
- [0036] 이후, 메모리 랜덤화 모듈은 기 설정된 시간 간격으로 상술한 220 단계 내지 250 단계를 반복적으로 수행하여, 공유 라이브러리를 메모리 공간 내에서 지속적으로 재배치한다.
- [0037] 한편, 도 2에 도시된 순서도에서는 상기 방법을 복수 개의 단계로 나누어 기재하였으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 추가되어 수행될 수 있다.
- [0038] 도 3은 본 발명의 일 실시예에 따른 메모리 랜덤화를 설명하기 위한 예시도이다.
- [0039] 도 3을 참조하면, 메모리 공간 내에 인젝트된 메모리 랜덤화 모듈(310)은 현재 활성화된 공유 라이브러리(320)를 임의의 주소 공간으로 복사하여 예비 공유 라이브러리(330)를 생성한다.
- [0040] 이후, 메모리 랜덤화 모듈(320)은 공유 라이브러리의 각 함수들에 대한 포인터가 예비 공유 라이브러리(330)의 함수 주소를 포인팅 하도록 포인터를 조정한 후, PLT(340)를 변경하여 예비 공유 라이브러리를 활성화시킨다(350).
- [0041] 이후, 메모리 랜덤화 모듈(320)은 도 3에 도시된 메모리 랜덤화 과정을 기 설정된 간격(예를 들어, 50ms)으로 반복 수행하여 메모리 공간 내에서 공유 라이브러리를 지속적으로 재배치시킬 수 있다.
- [0042] 이상에서 대표적인 실시예를 통하여 본 발명에 대하여 상세하게 설명하였으나, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 전술한 실시예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형

이 가능함을 이해할 것이다. 그러므로 본 발명의 권리범위는 설명된 실시예에 국한되어 정해져서는 안 되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

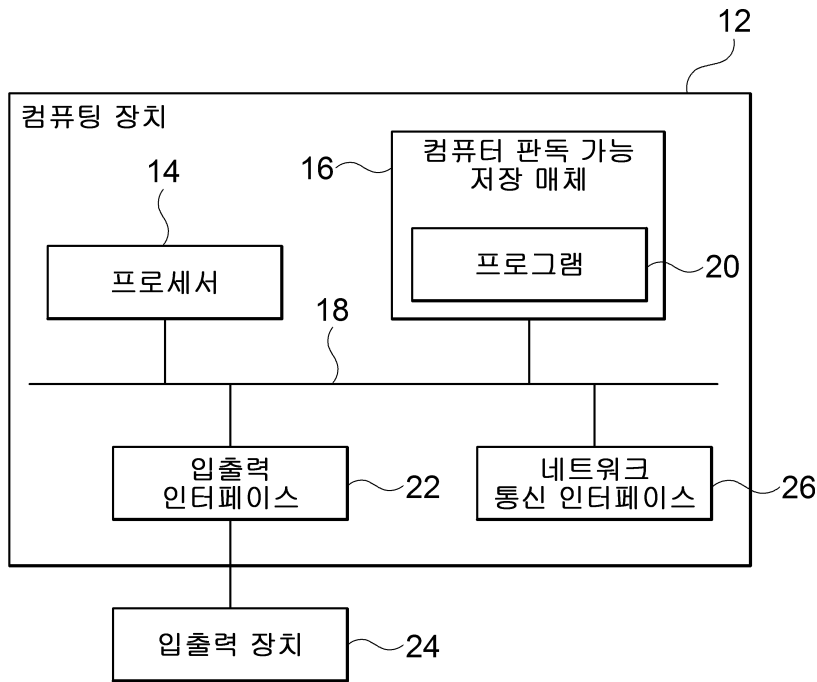
부호의 설명

- 10: 컴퓨팅 환경
- 12: 컴퓨팅 장치
- 14: 프로세서
- 16: 컴퓨터 판독 가능 저장 매체
- 18: 통신 버스
- 20: 프로그램
- 22: 입출력 인터페이스
- 24: 입출력 장치
- 26: 네트워크 통신 인터페이스

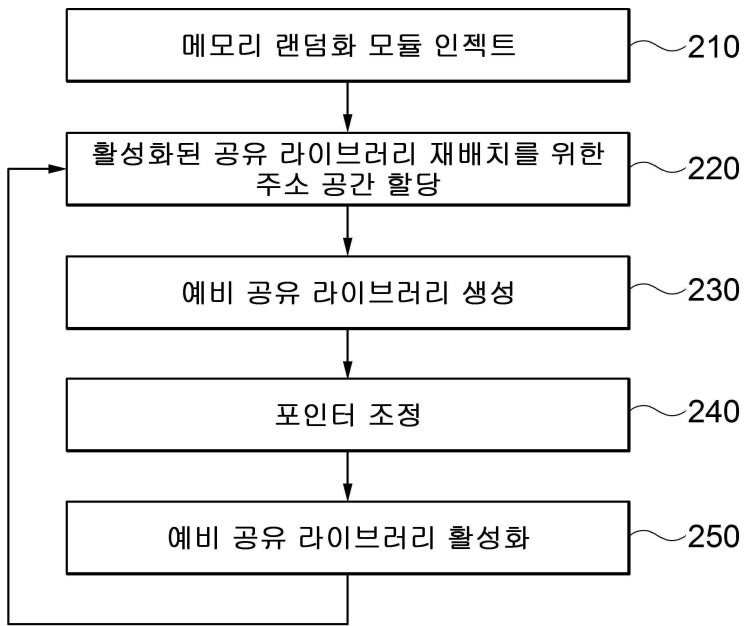
도면

도면1

10



도면2



도면3

