(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0092235
(43) 공개일자 2019년08월07일

(51) 국제특허분류(Int. Cl.)
G06F 21/51 (2013.01) *G06F 21/56* (2013.01)
(52) CPC특허분류
G06F 21/51 (2013.01)
G06F 21/56 (2013.01)
(21) 출원번호 10-2018-0133478
(22) 출원일자 2018년11월02일
심사청구일자 2018년11월02일
(30) 우선권주장
1020180011683 2018년01월30일 대한민국(KR)

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
박문찬
서울특별시 도봉구 쌍문2동 삼익아파트 113동 1506호
이동훈
서울특별시 종로구 사직로8길 34, 1404호(내수동, 경희궁의아침3단지아파트)
(74) 대리인
김홍석

전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **예측 불가능성에 기반한 효율적인 소프트웨어 제어흐름 무결성 검증 방법**

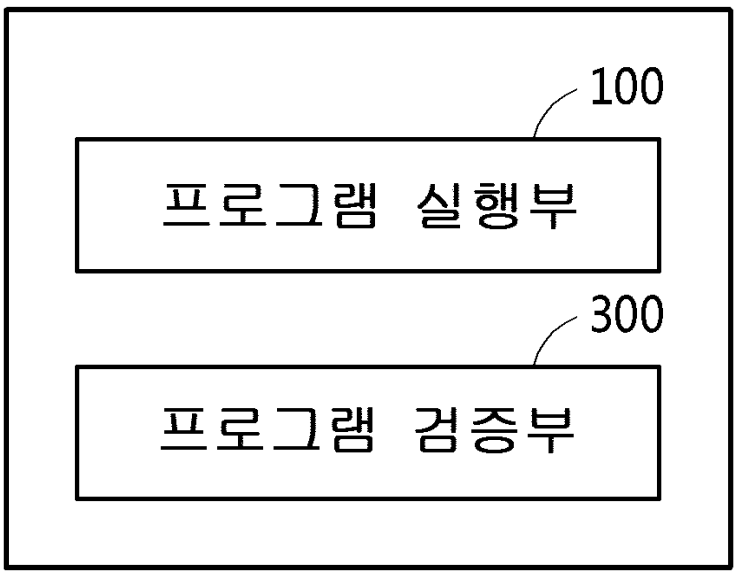
(57) 요 약

제어흐름 무결성 검증 방법이 개시된다. 상기 제어흐름 무결성 검증 방법은 타겟 프로그램을 실행하는 프로그램 실행부와 상기 타겟 프로그램의 제어흐름 무결성(Control-Flow Integrity, CFI)을 검증하는 프로그램 검증부를 포함하는 검증 장치에 의해 수행되고, (a) 상기 프로그램 실행부가 상기 타겟 프로그램을 실행한 후 상기 타겟

(뒷면에 계속)

대 표 도 - 도2

10



프로그램의 제어흐름 그래프(Control-Flow Graph, CFG)를 상기 프로그램 검증부로 송신하는 단계, (b) 상기 프로그램 실행부가, 상기 타겟 프로그램 내의 분기문이 발생한 경우 상기 프로그램 검증부로 제어흐름 무결성 검증 요청을 송신하는 단계, (c) 상기 제어흐름 무결성 검증 요청에 응답하여, 상기 프로그램 검증부가 소정의 확률에 기반하여 제어흐름 무결성 검증의 수행 여부를 결정하는 단계, (d) 상기 제어흐름 무결성 검증을 수행하기로 결정된 경우, 상기 프로그램 검증부가 상기 제어흐름 무결성 검증을 수행하는 단계, 및 (e) 상기 프로그램 검증부가 상기 제어흐름 무결성 검증의 검증 결과를 상기 프로그램 실행부로 송신하는 단계를 포함한다.

(52) CPC특허분류
G06F 2221/033 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 2017009643
부처명 과학기술정보통신부
연구관리전문기관 한국연구재단
연구사업명 (이공)중견연구자지원
연구과제명 향상된 제어흐름 무결성 검증 기법이 통합된 컴파일러 설계 및 구현
기 여 율 1/1
주관기관 고려대학교 산학협력단
연구기간 2017.03.01 ~ 2018.02.28

명 세 서

청구범위

청구항 1

타겟 프로그램을 실행하는 프로그램 실행부와 상기 타겟 프로그램의 제어흐름 무결성(Control-Flow Integrity, CFI)을 검증하는 프로그램 검증부를 포함하는 검증 장치에 의해 수행되는 제어흐름 무결성 검증 방법에 있어서,

(a) 상기 프로그램 실행부가 상기 타겟 프로그램을 실행한 후 상기 타겟 프로그램의 제어흐름 그래프(Control-Flow Graph, CFG)를 상기 프로그램 검증부로 송신하는 단계;

(b) 상기 프로그램 실행부가, 상기 타겟 프로그램 내의 분기문이 발생한 경우 상기 프로그램 검증부로 제어흐름 무결성 검증 요청을 송신하는 단계;

(c) 상기 제어흐름 무결성 검증 요청에 응답하여, 상기 프로그램 검증부가 소정의 확률에 기반하여 제어흐름 무결성 검증의 수행 여부를 결정하는 단계;

(d) 상기 제어흐름 무결성 검증을 수행하기로 결정된 경우, 상기 프로그램 검증부가 상기 제어흐름 무결성 검증을 수행하는 단계; 및

(e) 상기 프로그램 검증부가 상기 제어흐름 무결성 검증의 검증 결과를 상기 프로그램 실행부로 송신하는 단계를 포함하는 제어흐름 무결성 검증 방법.

청구항 2

제1항에 있어서,

상기 분기문은 콜(call), 점프(jmp), 또는 리턴(ret) 명령어이고,

상기 제어흐름 무결성 검증 방법은 (f) 상기 프로그램 실행부가 상기 검증 결과에 기초하여 상기 타겟 프로그램의 실행 중단 여부를 결정하는 단계를 더 포함하고,

상기 (b) 단계 내지 (f) 단계는 상기 타겟 프로그램의 실행 중 분기문이 발생할 때마다 반복적으로 수행되는,

제어흐름 무결성 검증 방법.

청구항 3

제2항에 있어서,

상기 (e) 단계는, 상기 검증 결과가 상기 제어흐름 검증의 실패를 의미하는 메시지인 경우 상기 타겟 프로그램의 실행을 중단하기로 결정하는,

제어흐름 무결성 검증 방법.

청구항 4

제3항에 있어서,

상기 (d) 단계는 상기 제어흐름 그래프(CFG)에 기초하여 상기 분기문에 의한 상기 제어흐름의 무결성 검증을 수행하는,

제어흐름 무결성 검증 방법.

청구항 5

제4항에 있어서,

상기 제어흐름 무결성 검증 방법은, 상기 (a) 단계 이전에,

상기 프로그램 검증부가 수정전 타겟 프로그램에 포함된 복수의 분기문들 각각의 이전 또는 이후에 상기 제어흐름 무결성 검증 요청을 수행하는 명령어를 삽입하여 상기 타겟 프로그램을 생성하는 단계를 더 포함하는,

제어흐름 무결성 검증 방법.

청구항 6

제4항에 있어서,

상기 제어흐름 무결성 검증 방법은, 상기 (a) 단계 이전에,

상기 프로그램 검증부가 상기 제어흐름 그래프(CFG)를 포함하지 않는 타겟 프로그램에 상기 제어흐름 그래프(CFG)를 삽입하여 상기 타겟 프로그램을 생성하는 단계를 더 포함하는,

제어흐름 무결성 검증 방법.

청구항 7

타겟 프로그램을 실행하는 프로그램 실행부와 상기 타겟 프로그램의 제어흐름 무결성(CFI)을 검증하는 프로그램 검증부를 포함하는 제어흐름 무결성 검증 장치에 있어서,

상기 프로그램 실행부는 상기 타겟 프로그램을 실행한 후 상기 타겟 프로그램의 제어흐름 그래프(Control-Flow Graph, CFG)를 상기 프로그램 검증부로 송신하고, 상기 타겟 프로그램 내의 분기문이 발생한 경우 상기 프로그램 검증부로 제어흐름 무결성 검증 요청을 송신하고,

상기 프로그램 검증부는 상기 제어흐름 무결성 검증 요청에 응답하여 소정의 확률에 기반하여 제어흐름 무결성 검증의 수행 여부를 결정하고, 상기 제어흐름 무결성 검증을 수행하기로 결정된 경우 상기 제어흐름 무결성 검증을 수행하고, 상기 제어흐름 무결성 검증의 검증 결과를 상기 프로그램 실행부로 송신하는,

제어흐름 무결성 검증 장치.

청구항 8

제7항에 있어서,

상기 분기문은 콜(call), 점프(jmp), 또는 리턴(ret) 명령어이고,

상기 프로그램 실행부는 상기 검증 결과에 기초하여 상기 타겟 프로그램의 실행 중단 여부를 결정하고,

상기 프로그램 실행부에 의한 상기 제어흐름 무결성 검증 요청의 송신 동작, 상기 프로그램 검증부에 의한 상기 제어흐름 무결성 검증의 수행 여부 결정 동작, 상기 제어흐름 무결성 검증을 수행하기로 결정된 경우 상기 프로그램 검증부에 의한 상기 제어흐름 무결성 검증의 수행 동작, 상기 프로그램 검증부에 의한 상기 검증 결과 송신 동작, 및 상기 프로그램 실행부에 의한 상기 타겟 프로그램의 실행 중단 여부 결정 동작은 상기 타겟 프로그램의 실행 중 분기문이 발생할 때마다 반복적으로 수행되는,

제어흐름 무결성 검증 장치.

청구항 9

제8항에 있어서,

상기 검증 결과가 상기 제어흐름 검증의 실패를 의미하는 메시지인 경우 상기 프로그램 실행부는 상기 타겟 프로그램의 실행을 중단하기로 결정하는,

제어흐름 무결성 검증 장치.

청구항 10

제9항에 있어서,

상기 프로그램 검증부는 상기 제어흐름 그래프(CFG)에 기초하여 상기 분기문에 의한 상기 제어흐름의 무결성 검증을 수행하는,

제어흐름 무결성 검증 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 소프트웨어의 제어흐름 무결성 검증 방법에 관한 것으로, 특히 예측 불가능한 확률에 기반하여 제어흐름의 무결성을 검증하여 악의적인 제어흐름 조작을 예방할 수 있는 소프트웨어 제어흐름 무결성 검증 방법에 관한 것이다.

배 경 기 술

[0003] 소프트웨어 또는 프로그램이 정상적으로 동작할 경우 정상 제어흐름 그래프(Control-Flow Graph, CFG) 상에 명시된 대로만 제어흐름(코드 실행 흐름)이 발생해야 한다. 그러나, 공격자가 소프트웨어 내의 취약점을 이용해서 정상적인 CFG 상에서는 발생할 수 없는 제어흐름을 발생시킬 수 있는데, 이를 제어흐름 조작 공격이라 한다. 대표적인 공격 기법으로 Buffer Overflow, Return-Oriented Programming(ROP) 등이 있다.

[0004] 소프트웨어의 제어흐름이 조작 가능하다는 것은 개발자가 의도하지 않은 임의의 기능을 공격자가 실행 가능하다는 것을 의미하고, 이는 결국 소프트웨어가 공격자의 의도대로 동작하도록 장악할 수 있다는 것을 의미한다.

[0005] 제어흐름 무결성(Control-Flow Integrity, CFI) 검증이란 소프트웨어 또는 프로그램의 제어흐름이 사전에 정의된 CFG를 따라 실행되는지를 검증하는 기술을 의미하며, 다양한 형태의 제어흐름 조작 공격을 근본적으로 막기 위한 연구가 수행되고 있다. 도 1과 같이 CFI 검증 메커니즘은 소프트웨어의 제어흐름(도 1의 Control-flow in target software)이 정상 CFG 정보(도 1의 Original CFG DB)에 명시되어 있다면 이를 허용하고, 명시되어 있지 않다면 이를 탐지한다.

[0006] CFI 검증에서는 다른 소프트웨어 보호 기법과 같이 보안성(제어흐름 조작 공격 저항성)과 실용성(적은 퍼포먼스 저하)을 모두 고려해야 한다. 그러나, 현재 제안된 CFI 기법의 경우 보안성과 실용성 중 어느 하나에 초점을 맞추고 있기 때문에 소프트웨어에 실적용 하기에는 어려움이 따른다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 대한민국 공개특허 제2017-0079858호 (2017.07.10. 공개)

(특허문헌 0002) 대한민국 등록특허 제1575021호 (2015.12.01. 등록)

(특허문헌 0003) 대한민국 공개특허 제2009-0066059호 (2009.06.23. 공개)

발명의 내용

해결하려는 과제

[0009] 본 발명이 이루고자 하는 기술적인 과제는 예측 불가능한 확률에 기반한 효율적인 소프트웨어 제어흐름 무결성 검증 방법을 제공하는 것이다.

과제의 해결 수단

[0011] 본 발명의 실시 예에 따른 제어흐름 무결성 검증 방법은 타겟 프로그램을 실행하는 프로그램 실행부와 상기 타겟 프로그램의 제어흐름 무결성(Control-Flow Integrity, CFI)을 검증하는 프로그램 검증부를 포함하는 검증 장치에 의해 수행되고, (a) 상기 프로그램 실행부가 상기 타겟 프로그램을 실행한 후 상기 타겟 프로그램의 제어흐름 그래프(Control-Flow Graph, CFG)를 상기 프로그램 검증부로 송신하는 단계, (b) 상기 프로그램 실행부가, 상기 타겟 프로그램 내의 분기문이 발생한 경우 상기 프로그램 검증부로 제어흐름 무결성 검증 요청을 송신하는 단계, (c) 상기 제어흐름 무결성 검증 요청에 응답하여, 상기 프로그램 검증부가 소정의 확률에 기반하여 제어흐름 무결성 검증의 수행 여부를 결정하는 단계, (d) 상기 제어흐름 무결성 검증을 수행하기로 결정된 경우, 상기 프로그램 검증부가 상기 제어흐름 무결성 검증을 수행하는 단계, 및 (e) 상기 프로그램 검증부가 상기 제어흐름 무결성 검증의 검증 결과를 상기 프로그램 실행부로 송신하는 단계를 포함한다.

[0012] 또한, 본 발명의 실시 예에 따른 제어흐름 무결성 검증 장치는 타겟 프로그램을 실행하는 프로그램 실행부와 상기 타겟 프로그램의 제어흐름 무결성(CFI)을 검증하는 프로그램 검증부를 포함하고, 상기 프로그램 실행부는 상기 타겟 프로그램을 실행한 후 상기 타겟 프로그램의 제어흐름 그래프(Control-Flow Graph, CFG)를 상기 프로그램 검증부로 송신하고, 상기 타겟 프로그램 내의 분기문이 발생한 경우 상기 프로그램 검증부로 제어흐름 무결성 검증 요청을 송신하고, 상기 프로그램 검증부는 상기 제어흐름 무결성 검증 요청에 응답하여 소정의 확률에 기반하여 제어흐름 무결성 검증의 수행 여부를 결정하고, 상기 제어흐름 무결성 검증을 수행하기로 결정된 경우 상기 제어흐름 무결성 검증을 수행하고, 상기 제어흐름 무결성 검증의 검증 결과를 상기 프로그램 실행부로 송신한다.

발명의 효과

[0014] 본 발명의 실시 예에 따른 소프트웨어 제어흐름 검증 방법에 의할 경우, 정밀한 CFG를 이용하여 보안성을 높일 수 있다.

[0015] 또한, 본 발명에 의할 경우 예측 불가능한 확률에 기반하여 검증 시점을 결정하여 검증의 횟수를 감소시킴으로써 소프트웨어의 성능 저하를 방지하여 실용성을 높일 수 있다.

도면의 간단한 설명

[0017] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 제어흐름 무결성 검증의 개요를 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시 예에 의한 제어흐름 무결성 검증 장치의 기능 블럭도이다.

도 3은 도 2에 도시된 검증 장치에서 수행되는 제어흐름 무결성 검증 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시 예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시 예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시 예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시 예들에 한정되지 않는다.

[0019] 본 발명의 개념에 따른 실시 예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시 예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시 예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변

경, 균등물, 또는 대체물을 포함한다.

[0020] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0021] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0022] 본 명세서에서 사용한 용어는 단지 특정한 실시 예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0023] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0024] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시 예들을 상세히 설명한다. 그러나, 특허출원의 범위가 이러한 실시 예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

[0026] 도 2는 본 발명의 일 실시 예에 의한 제어흐름 무결성 검증 장치의 기능 블럭도이다.

[0027] 도 2를 참조하면, 제어흐름 무결성 검증 장치, 무결성 검증 장치 등으로 명명될 수도 있는 검증 장치(10)는 프로그램 실행부(100)와 프로그램 검증부(300)를 포함한다. 검증 장치(10)는 타겟 프로그램(또는 타겟 소프트웨어)을 실행하고 실행 중인 타겟 프로그램의 제어흐름 무결성(Control-Flow Integrity, CFI)을 검증할 수 있다. 본 발명이 적용되는 환경은 상시적으로 소프트웨어의 정상적인 동작이 요구되는 소프트웨어 실행 환경일 수 있다. 예컨대, 소프트웨어의 오작동이 심각한 피해로 이어질 수 있는 자율주행 자동차, 발전소와 같은 cirtical infrastructure, 또는 모든 하드웨어가 소프트웨어로 직접 제어되는 CPS(Cyber-Physical System) 등의 환경이 될 수 있다.

[0028] 프로그램 실행부(100)는 검증 대상인 타겟 프로그램을 실행할 수 있다. 구체적으로, 프로그램 실행부(100)는 상기 타겟 프로그램을 실행하고, 상기 타겟 프로그램의 제어흐름 그래프(Control-Flow Graph, CFG)를 프로그램 검증부(300)로 송신할 수 있다. 프로그램 실행부(100)는 상기 CFG를 송신함으로써 상기 타겟 프로그램이 검증 대상 프로그램임을 프로그램 검증부(300)에 알릴 수 있다. 실시 예에 따라, 상기 제어흐름 그래프(CFG)는 상기 타겟 프로그램의 실행과 동시에 프로그램 검증부(300)로 송신되는 것으로 이해될 수도 있다.

[0029] 또한, 프로그램 실행부(100)는 상기 타겟 프로그램의 실행 중에 발생하는(또는 실행되는) 분기문(또는 분기 명령문)의 실행 전, 실행 후, 또는 실행과 동시에 상기 분기문에 따른 타겟 프로그램의 제어흐름 무결성의 검증 요청을 프로그램 검증부(300)로 송신할 수 있다. 상기 타겟 프로그램에는 복수개의 분기문이 존재할 수 있으며, 이러한 분기문이 발생할 때마다 프로그램 실행부(100)는 검증 요청을 송신할 수 있다.

[0030] 프로그램 실행부(100)는 검증 요청에 대한 프로그램 검증부(300)의 응답에 따라 상기 타겟 프로그램의 실행을 중단할 수 있다. 즉, 검증 요청에 대한 응답이 검증 성공을 의미하는 메시지이거나 검증을 수행하지 않았음을 의미하는 메시지인 경우 타겟 프로그램의 실행을 계속하고, 검증 요청에 대한 응답이 검증 실패를 의미하는 메

시지인 경우 타겟 프로그램의 실행을 중단할 수 있다.

[0031] 실시 예에 따라, 프로그램 실행부(100)가 실행하는 타겟 프로그램은 프로그램 검증부(300)에 의해 컴파일된(또는 수정된) 프로그램, 즉 컴파일된 타겟 프로그램(또는 수정된 타겟 프로그램)을 의미할 수 있다.

[0033] 프로그램 검증부(300)는 프로그램 실행부(100)로부터 제어흐름 그래프(CFG)를 수신한다. 수신된 제어흐름 그래프(CFG)는 별도의 저장 공간에 저장될 수 있다. 또한, 프로그램 검증부(300)는 프로그램 실행부(100)의 검증 요청에 응답하여 제어흐름을 검증하고, 검증 결과를 프로그램 실행부(100)로 송신할 수 있다.

[0034] 실시 예에 따라, 프로그램 검증부(300)는 상기 타겟 프로그램을 컴파일하거나 수정할 수 있다. 즉, 상기 타겟 프로그램이 컴파일되지 않았다면, 상기 타겟 프로그램을 컴파일할 수 있고, 컴파일 도중 또는 컴파일을 완료한 후에 상기 타겟 프로그램에 포함된 분기문의 전 또는 이후에 검증 요청을 수행하는 명령어를 삽입할 수 있다. 상기 타겟 프로그램이 컴파일된 프로그램이라면, 프로그램 검증부(300)는 상기 타겟 프로그램에 포함된 분기문의 전 또는 이후에 검증 요청을 수행하는 명령어를 삽입할 수 있다.

[0036] 도 3은 도 2에 도시된 검증 장치에서 수행되는 제어흐름 무결성 검증 방법을 설명하기 위한 흐름도이다.

[0037] 도 2와 도 3을 참조하면, 우선 프로그램 실행부(100)는 검증 장치(10)의 일부가 될 수 있는 소정의 저장 장치에 저장된 타겟 프로그램을 로드하여 상기 타겟 프로그램을 실행할 수 있다. 이때, 프로그램 실행부(100)는 상기 타겟 프로그램에 포함되어 있거나 상기 타겟 프로그램과 함께 저장되어 있는 상기 타겟 프로그램의 제어흐름 그래프(CFG)를 프로그램 검증부(300)로 송신할 수 있다.

[0038] 프로그램 검증부(300)는 프로그램 실행부(100)로부터 수신받은 CFG를 별도의 저장 공간에 저장함으로써, 상기 타겟 프로그램의 구동 과정에서 공격자의 공격에 의한 CFG의 조작에 영향을 받지 않고 상기 타겟 프로그램의 제어 흐름 무결성을 검증할 수 있다.

[0039] 상기 타겟 프로그램의 실행 중 분기문(또는 분기 명령어)이 발생한 경우(또는 출현한 경우), 프로그램 실행부(100)는 프로그램 검증부(300)로 제어흐름 무결성 검증 요청을 송신할 수 있다. 상기 제어흐름 무결성 검증 요청은 발생한 상기 분기문이 실행되기 전, 실행된 후, 또는 실행과 동시에 송신될 수 있다. 상기 타겟 프로그램의 실행은 상기 검증 요청에 대한 응답이 수신될 때까지 중단될 수 있으나, 실시 예에 따라 상기 타겟 프로그램은 상기 검증 요청이 송신된 이후에도 중단되지 않을 수도 있다. 상기 타겟 프로그램이 자율 주행 등과 같은 실시간으로 중단 없이 수행되어야 할 프로그램일 수도 있기 때문이다.

[0040] 제어흐름 무결성 검증 요청을 수신한 프로그램 검증부(300)는 제어흐름 무결성 검증의 수행 여부를 결정한다. 구체적으로, 프로그램 검증부(300)는 소정의 확률(p)로 "1"(또는 "0")을 출력하고 그 외에는(즉, p-1의 확률로) "0"(또는 "1")을 출력하는 함수를 이용하여 검증의 수행 여부를 결정할 수 있다. 즉, 상기 함수가 "1"(또는 "0")을 출력하는 경우 검증을 수행하고, 상기 함수가 "0"(또는 "1")을 출력하는 경우 검증을 수행하지 않을 수 있다. 상기 소정의 확률(p)은 미리 정해진 상수일 수 있으나, 본 발명이 이에 제한되는 것은 아니며, 상기 소정의 확률(p)은 타겟 프로그램의 종류, 길이, 분기문의 개수 등에 따라 변할 수도 있다. 또한, 타겟 프로그램의 보안성을 강조하기 위하여 큰 값을 갖는 p가 사용될 수 있고, 타겟 프로그램의 실용성(성능 저하 방지)를 위하여 작은 값을 갖는 p가 사용될 수 있다.

[0041] 프로그램 검증부(300)가 검증 수행을 하지 않기로 결정한 경우, 즉 상기 함수의 출력 값이 "0"(또는 "1)인 경우, 프로그램 검증부(300)는 검증 결과, 즉 검증을 수행하지 않았음을 의미하는 메시지 또는 검증 성공을 의미하는 메시지를 프로그램 실행부(100)로 송신할 수 있다.

[0042] 프로그램 검증부(300)가 검증을 수행하기로 결정한 경우, 즉 상기 함수의 출력 값이 "1"(또는 "0"인 경우, 프로그램 검증부(300)는 미리 수신된 제어흐름 그래프(CFG)를 이용하여 분기문에 의한 제어흐름의 무결성 여부를 검증할 수 있다. 즉, 분기문에 의한 제어흐름 이동이 제어흐름 그래프(CFG)에 미리 정의된 정상적인 제어흐름 이동에 해당하는 경우 검증은 성공하고, 분기문에 의한 제어흐름 이동이 제어흐름 그래프(CFG)에 미리 정의된 정상적인 제어흐름 이동에 해당하지 않는 경우 검증은 실패하게 된다. 검증 결과(검증 성공을 의미하는 메시지 또는 검증 실패를 의미하는 메시지)는 프로그램 실행부(100)로 송신된다. 프로그램 검증부(300)에 의한 제어흐름 무결성 검증의 예로, 분기문이 콜(call, 함수 호출) 명령어인 경우 스택(또는 스택 메모리)에 저장된 복귀 주소(return address, 리턴 어드레스)와 제어흐름 그래프(CFG)에 명시된 복귀 주소의 동일성 여부에 따라 검증 성공

여부가 결정되거나 콜 명령어에 따라 호출되는 함수(또는 함수의 어드레스)와 제어흐름 그래프(CFG)에 명시된 호출되는 함수(또는 호출되는 함수의 어드레스)의 동일성 여부에 따라 검증 성공 여부가 결정될 수 있고, 분기문이 점프(jump) 명령어인 경우 점프할(이동할) 주소(address)와 제어흐름 그래프(CFG)에 명시된 주소의 동일성 여부에 따라 검증 성공 여부가 결정될 수 있고, 분기문이 리턴(return)인 경우 리턴 명령에 따라 복귀할 리턴 어드레스와 제어흐름 그래프(CFG)에 명시된 리턴 어드레스의 동일성 여부에 따라 검증 성공 여부가 결정될 수 있다.

[0043] 검증 결과를 수신한 프로그램 실행부(100)는 검증 결과에 따라 타겟 프로그램의 계속 실행 여부를 결정할 수 있다. 구체적으로, 검증 결과가 검증 성공을 의미하는 메시지(검증을 수행하지 않았음을 의미하는 메시지를 포함함)인 경우 타겟 프로그램은 계속적으로 수행될 수 있으나, 검증 결과가 검증 실패를 의미하는 메시지인 경우 타겟 프로그램은 중단될 수 있다.

[0044] 상술한 S200 단계 내지 S600 단계는 타겟 프로그램에서 분기문이 실행될 때마다 반복적으로 수행될 수 있다.

[0046] 다른 실시 예로, 제어흐름 무결성 검증 방법은 S100 단계 이전에, 프로그램 검증부(300)가 타겟 프로그램에 포함된 분기문의 전 또는 후에 무결성 검증 요청을 수행하는 명령어를 삽입하여 검증 요청 명령어가 삽입된 타겟 프로그램(즉, 수정된 타겟 프로그램)을 생성하는 단계 및/또는 프로그램 검증부(300)가 상기 타겟 프로그램의 제어흐름 그래프를 생성하는 단계를 더 포함할 수 있다. 또한, 상기 수정된 타겟 프로그램의 생성 동작과 상기 타겟 프로그램의 구동 동작은 상이한 시간에 상이한 주체에 의해 수행될 수도 있다.

[0048] 또 다른 실시 예로, 제어흐름 무결성 검증 방법은 S100 단계 이전에, 프로그램 검증부(300)가 컴파일되지 않은 타겟 프로그램을 컴파일함으로써 컴파일된 타겟 프로그램을 생성하는 단계를 더 포함할 수 있다. 이때, 프로그램 검증부(300)는 상기 타겟 프로그램의 제어흐름 그래프(control-flow graph, CFG)를 생성하고, 생성된 CFG를 상기 컴파일된 타겟 프로그램 내에 삽입하거나, 상기 컴파일된 타겟 프로그램과 함께 저장할 수 있다.

[0049] 또한, 프로그램 검증부(300)는 상기 타겟 프로그램에 포함된 복수의 분기 명령어들의 이전 또는 이후에 프로그램 검증부(300)에 제어흐름 무결성 검증을 요청하는 명령어를 삽입할 수 있다. 무결성 검증을 요청하는 명령어의 삽입은 컴파일 과정에서 수행되거나 컴파일이 완료된 후에 수행될 수 있다. 따라서, 상기 컴파일된 타겟 프로그램에는 분기 명령어 이전 또는 이후에 존재하는 검증 요청 명령어가 삽입될 수 있으며, 프로그램 실행부(100)는 상기 컴파일된 타겟 프로그램을 실행되는 동안 분기 명령어가 수행되기 전 또는 이후에 프로그램 검증부(300)에 제어흐름 무결성 검증을 요청할 수 있다. 상기 분기문은 콜(call), 점프(jmp), 및/또는 리턴(ret) 명령어일 수 있다.

[0051] 제어흐름 조작 공격은 연속적인 제어흐름 조작을 통해 공격이 이루어진다. 이때 제어흐름 조작을 통해 수행되는 코드 조각들은 논리적으로 아주 강하게 결합되어 있기 때문에 공격자가 조작한 모든 코드 조각들이 연속적으로 실행되어야만 최종적으로 공격자가 의도한 목적을 달성할 수 있다. 그러나 이는 역으로, 연속적으로 발생하는 제어흐름 조작 중에서 어느 하나라도 제어흐름 무결성(CFI) 검사를 통해 탐지할 수 있다면 공격자가 의도한 행위를 무산시킬 수 있다는 것을 의미한다. 이러한 관점에서 모든 제어흐름 이동을 검사하기 보다는 선택적으로 제어흐름 이동을 검사하는 것이 보안성을 최대한 유지하면서도 실용성을 향상시킬 수 있는 방안이다. 선택적 검사를 수행할 때, 공격자가 검사를 우회할 수 없도록 하기 위해서는 검사 시점이 "예측 불가능"한 성질을 가져야 한다. 검사 시점을 공격자가 예측할 수 있다면 각 제어흐름 조작마다 (1-p)의 확률로 검증을 통과할 수 있다. 즉, 공격자가 최종적으로 의도한 행위를 수행하기 위해 총 m 번의 제어흐름 조작이 필요하다면 공격 성공 확률은 $(1-p)^m$이 된다. 예를 들어, p=1/2이고 m=5라면 공격 성공 확률은 약 3%에 불과하며, m=10이라면 공격 성공확률은 0.1% 미만이 된다.

[0053] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 집합으로 구현될 수 있다. 예를 들어, 실시 예들에서 설명된 장치 및 구성 요소는, 예를 들어, 프로세서, 콘트롤러, ALU(Arithmetic Logic Unit), 디지털 신호 프로세서(Digital Signal Processor), 마이크로컴퓨터, FPA(Field Programmable array), PLU(Programmable Logic Unit), 마이크로프로세서, 또는 명령

(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(Operation System, OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술 분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(Processing Element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(Parallel Processor)와 같은, 다른 처리 구성(Processing Configuration)도 가능하다.

[0054] 소프트웨어는 컴퓨터 프로그램(Computer Program), 코드(Code), 명령(Instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(Collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성 요소(Component), 물리적 장치, 가상 장치(Virtual Equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(Signal Wave)에 영구적으로, 또는 일시적으로 구체화(Embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0055] 실시 예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 좋ㅂ하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시 예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM, DVD와 같은 광기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-optical Media), 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시 예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0057] 본 발명은 도면에 도시된 실시 예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성 요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성 요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

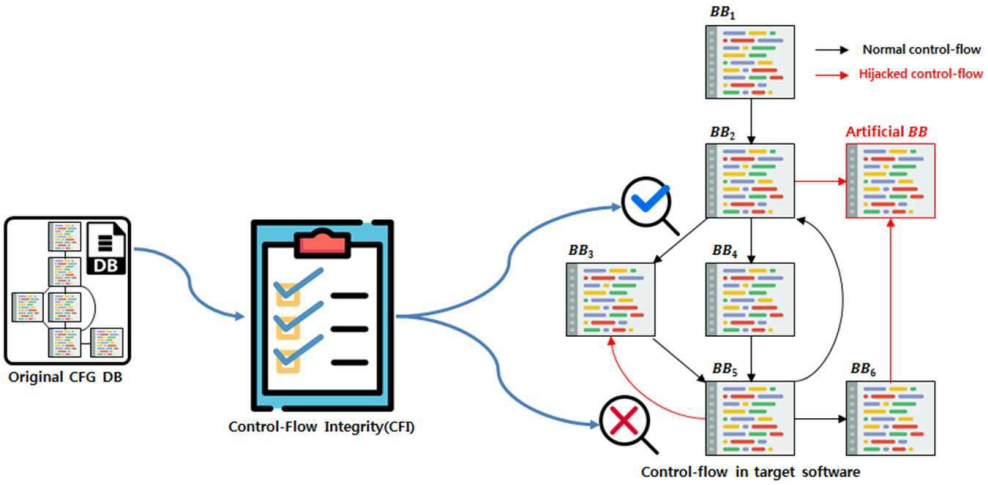
[0059] 10 : 검증 장치

100 : 프로그램 실행부

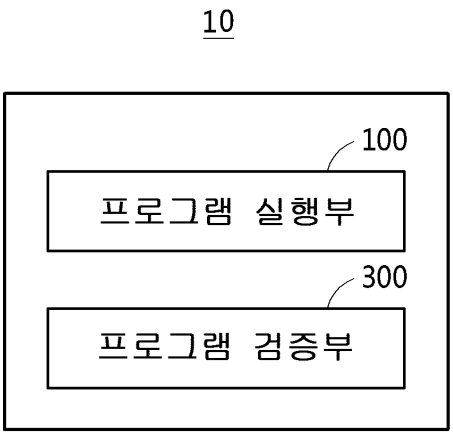
300 : 프로그램 검증부

도면

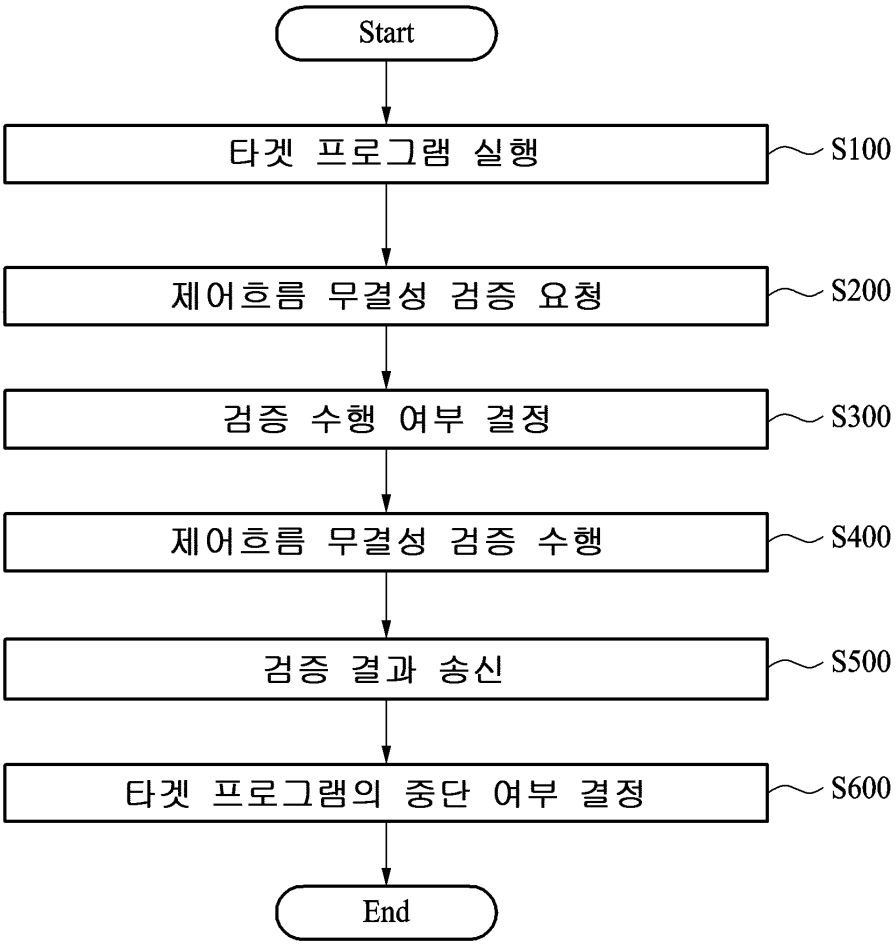
도면1


BB_1
Normal control-flow
Hijacked control-flow
BB_2
Artificial BB
BB_3
BB_4
BB_5
BB_6
DB
Original CFG DB
Control-Flow Integrity(CFI)
Control-flow in target software

도면2


10
100
프로그램 실행부
300
프로그램 검증부

도면3


Start
타겟 프로그램 실행
S100
제어흐름 무결성 검증 요청
S200
검증 수행 여부 결정
S300
제어흐름 무결성 검증 수행
S400
검증 결과 송신
S500
타겟 프로그램의 중단 여부 결정
S600
End