(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년09월24일
(11) 등록번호 10-2305386
(24) 등록일자 2021년09월16일

(51) 국제특허분류(Int. Cl.)
*G06F 11/36* (2006.01)
(52) CPC특허분류
*G06F 11/3664* (2013.01)
*G06F 11/3684* (2013.01)
(21) 출원번호 10-2021-0078620
(22) 출원일자 2021년06월17일
심사청구일자 2021년06월17일
(56) 선행기술조사문헌
KR102209676 B1*
*(뒷면에 계속)*

(73) 특허권자
**세종대학교산학협력단**
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
**윤주범**
서울특별시 송파구 충민로4길 19, 704동 401호(장지동, 송파파인타운7단지)
**유지현**
서울특별시 광진구 군자로3길 18-2, 101호 (화양동)
*(뒷면에 계속)*
(74) 대리인
**두호특허법인**

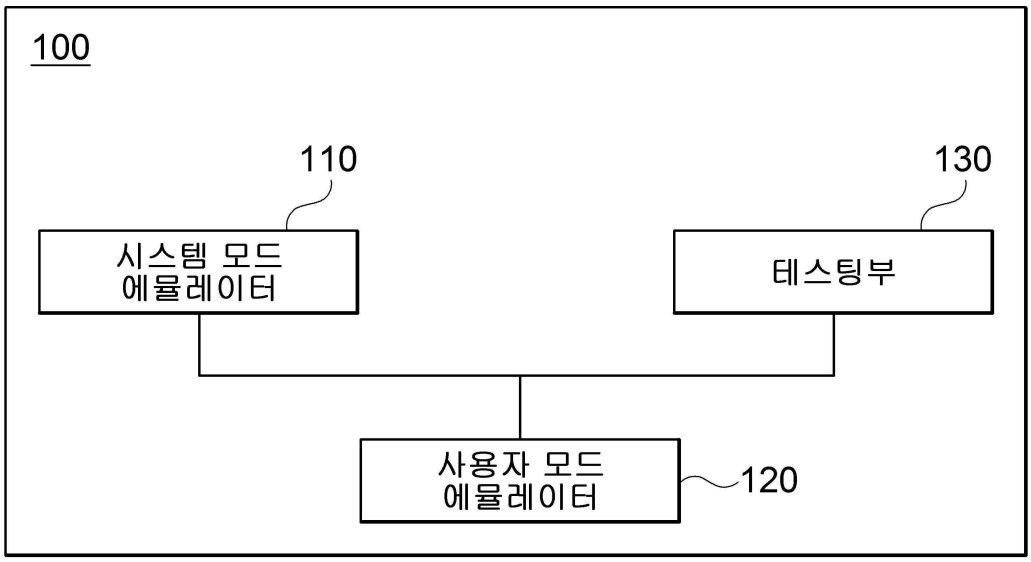
전체 청구항 수 : 총 10 항 심사관 : 김계준

(54) 발명의 명칭 **펌웨어 퍼징 장치 및 방법**

## (57) 요 약

펌웨어 퍼징 장치 및 방법이 개시된다. 일 실시예에 따르면 펌웨어 퍼징 장치는 휴리스틱(heuristics) 기법에 기초하여 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 시스템 모드 에뮬레이터; 상기 시스템 모드 에뮬레이션 환경에서 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 사용자 모드 에뮬레이터; 및 상기 대상 프로세스 내 포함된 키워드에 기초하여 테스트 케이스를 생성하고, 상기 테스트 케이스를 이용하여 상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 변이 기반 퍼징(fuzzing)을 실행하는 테스팅부를 포함한다.

*대 표 도* - 도1



(52) CPC특허분류
***G06F 11/3688*** (2013.01)
***G06F 9/45504*** (2013.01)
***G06F 9/45533*** (2013.01)
***G06F 9/45558*** (2013.01)
*G06F 2009/45562* (2013.01)
*G06F 2009/45591* (2019.08)

(72) 발명자
**김주환**
서울특별시 광진구 군자로 175-2, 304호(군자동)
**이영우**
경기도 구리시 경춘로288번길 39, 마동 410호(수택동)

(56) 선행기술조사문헌
JP2019091411 A*
JP2017076412 A*
JP2019220141 A*
KR101972825 B1
*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업
과제고유번호 1711126109
과제번호 2018-0-01423-004
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 대학ICT연구센터육성지원사업
연구과제명 지능형 비행로봇 융합기술 연구
기 여 율 1/2
과제수행기관명 세종대학교 산학협력단
연구기간 2021.01.01 ~ 2021.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711126138
과제번호 2020-0-01602-002
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성(R&D)
연구과제명 지능형 사이버 위협 대응 기술 개발 및 인력양성
기 여 율 1/2
과제수행기관명 숭실대학교 산학협력단
연구기간 2021.01.01 ~ 2021.12.31
공지예외적용 : 있음

# 명 세 서

## 청구범위

### 청구항 1

휴리스틱(heuristics) 기법에 기초하여 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 시스템 모드 에뮬레이터;

상기 시스템 모드 에뮬레이션 환경에서 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 사용자 모드 에뮬레이터;

동적 기호 실행(concolic execution)을 통해 분기 조건을 해결하는 키워드를 상기 대상 프로세스에서 검출하는 검출부; 및

상기 검출된 키워드에 기초하여 특정 형식을 갖춘 테스트 케이스를 생성하고, 상기 테스트 케이스를 이용하여 상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 변이 기반 퍼징(fuzzing)을 실행하는 테스팅부를 포함하는, 펌웨어 퍼징 장치.

### 청구항 2

청구항 1에 있어서,

상기 시스템 모드 에뮬레이터는,

상기 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스를 반복 실행하고, 반복 실행하여 획득된 결과 중 상기 시스템 모드 에뮬레이션 환경의 제공에 성공한 케이스에 기초하여 상기 시스템 모드 에뮬레이션 환경을 제공하는, 펌웨어 퍼징 장치.

### 청구항 3

청구항 1에 있어서,

상기 대상 프로세스는,

사용자로부터 입력을 요청하는 프로세스인, 펌웨어 퍼징 장치.

### 청구항 4

청구항 1에 있어서,

상기 특정 형식은,

상기 대상 프로세스의 실행에 사용되는 데이터가 요구하는 형식인, 펌웨어 퍼징 장치.

### 청구항 5

청구항 1에 있어서,

상기 특정 형식은,

상기 대상 프로세스에 입력되는 데이터가 요구하는 형식인, 펌웨어 퍼징 장치.

**청구항 6**

삭제

**청구항 7**

휴리스틱(heuristics) 기법에 기초하여 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 단계;

상기 시스템 모드 에뮬레이션 환경에서 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 단계;

동적 기호 실행(concolic execution)을 통해 분기 조건을 해결하는 키워드를 상기 대상 프로세스에서 검출하는 단계;

상기 검출된 키워드를 이용하여 특정 형식을 갖춘 테스트 케이스를 생성하는 단계; 및

상기 테스트 케이스를 이용하여 상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 변이 기반 퍼징(fuzzing)을 실행하는 단계를 포함하는, 펌웨어 퍼징 방법.

**청구항 8**

청구항 7에 있어서,

상기 시스템 모드 에뮬레이션 환경을 제공하는 단계는,

상기 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스를 반복 실행하고, 반복 실행하여 획득된 결과 중 상기 시스템 모드 에뮬레이션 환경의 제공에 성공한 케이스에 기초하여 상기 시스템 모드 에뮬레이션 환경을 제공하는, 펌웨어 퍼징 방법.

**청구항 9**

청구항 7에 있어서,

상기 대상 프로세스는,

사용자로부터 입력을 요청하는 프로세스인, 펌웨어 퍼징 방법.

**청구항 10**

청구항 7에 있어서,

상기 특정 형식은,

상기 대상 프로세스의 실행에 사용되는 데이터가 요구하는 형식인, 펌웨어 퍼징 방법.

**청구항 11**

청구항 7에 있어서,

상기 특정 형식은,

상기 대상 프로세스에 입력되는 데이터가 요구하는 형식인, 펌웨어 퍼징 방법.

**청구항 12**

삭제

## 발명의 설명

### 기 술 분 야

[0001] 개시되는 실시예들은 펌웨어(firmware)에 대한 변이 기반 퍼징(fuzzing)을 수행하는 기술과 관련된다.

### 배 경 기 술

[0002] 해가 갈수록 다양한 임베디드 장치들이 일상의 모든 부분에 자리잡고 있다. 임베디드 장치의 수준은 양적, 질적으로 자율 주행, 의료 산업 등 민감한 분야에 적용될 정도로 발전하고 있다.

[0003] 하지만 임베디드 장치의 발전은 기기의 성능과 개발 성능에 치중된 나머지, 임베디드 장치에 대한 보안은 상대적으로 발전이 더딘 상황이다.

[0004] 이 때문에, 임베디드 장치에 내장된 펌웨어는 악의의 공격자에게 있어 쉽게 공격 대상의 타겟이 된다. 이에 대응하여, 일각에서는 임베디드 장치 내부의 펌웨어를 자동으로 테스트하는 여러 연구를 진행하고 있다. 예컨대, 첫째로는 에뮬레이션의 안정성 및 신뢰성 연구가, 둘째로는 이전 소프트웨어에서 성공적이었던 테스팅 기법을 펌웨어에 적용하는 연구가 이루어지고 있다.

[0005] 이러한 연구의 성과로서 구현된 FirmAFL은 자동 에뮬레이션 도구와 커버리지 기반 퍼징 도구를 결합하여 에뮬레이션과 퍼징을 자동으로 수행하는 도구로, 증강 프로세스 에뮬레이션을 통해 에뮬레이션의 불안정성과 퍼징 처리량을 개선하였다. 하지만 대부분의 임베디드 장치를 대상으로 한 퍼저와 마찬가지로 FirmAFL 역시 소수의 펌웨어만 호환되는 가공되지 않은 QEMU(Quick EMUlator)를 기반으로 에뮬레이션을 수행하기 때문에 에뮬레이션 성공률이 약 12 %로 매우 낮다.

[0006] 또한, 펌웨어 프로그램은 웹 어플리케이션을 통해 사용자의 입/출력 통신을 수행한다. 이때, 웹 어플리케이션은 특정 형식을 갖는 패킷 형태로 데이터를 주고받기 때문에 무작위 변이를 통해 입력값을 생성하는 기존 퍼저는 효율성이 떨어질 수 있다. 이를 보완하기 위해서는 프로그램에서 입력 형식을 수동적으로 추출하는 과정이 필요하지만 굉장히 많은 인력과 비용이 요구된다.

[0007] 이를 해결하기 위해서는 임베디드 장치에 대한 높은 에뮬레이션 성공률, 자동화된 입력 형식 구성으로 퍼저의 효율성을 높이는 시스템이 필요하며, 특히 기존의 소프트웨어에서 프로그램의 키워드 및 분기에서의 입력 값을 획득하기 위해 이용되어 온 기호 실행 기법을 퍼징 기법과 결합하는 방안을 고찰할 필요가 있다.

### 선행기술문헌

#### 특허문헌

[0008] (특허문헌 0001) 대한민국 등록특허공보 제10-1972825호(2019.04.22. 등록)

## 발명의 내용

### 해결하려는 과제

[0009] 개시되는 실시예들은 펌웨어(firmware)에 대한 변이 기반 퍼징(fuzzing)을 수행하기 위한 장치 및 방법을 제공하기 위한 것이다.

### 과제의 해결 수단

[0010] 일 실시예에 따른 펌웨어 퍼징 장치는 휴리스틱(heuristics) 기법에 기초하여 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 시스템 모드 에뮬레이터; 상기 시스템 모드 에뮬레이션 환경에서 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 사용자 모드 에뮬레이터; 및 상기 대상 프로세스 내 포함된 키워드에 기초하여 특정 형식을 갖춘 테스트 케이스를 생성하고, 상기 테스트 케이스를 이용하여 상기 사용자 모드 에뮬레이션 환경에서 상기 대상

프로세스에 변이 기반 퍼징(fuzzing)을 실행하는 테스팅부를 포함한다.

[0011] 상기 시스템 모드 에뮬레이터는, 상기 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스를 반복 실행하고, 반복 실행하여 획득된 결과 중 상기 시스템 모드 에뮬레이션 환경의 제공에 성공한 케이스에 기초하여 상기 시스템 모드 에뮬레이션 환경을 제공할 수 있다.

[0012] 상기 대상 프로세스는, 사용자로부터 입력을 요청하는 프로세스일 수 있다.

[0013] 상기 특정 형식은, 상기 대상 프로세스의 실행에 사용되는 데이터가 요구하는 형식일 수 있다.

[0014] 상기 특정 형식은, 상기 대상 프로세스에 입력되는 데이터가 요구하는 형식일 수 있다.

[0015] 일 실시예에 따른 펌웨어 퍼징 장치는 동적 기호 실행(concolic execution)을 통해 분기 조건을 해결하는 키워드를 상기 대상 프로세스에서 검출하는 검출부를 더 포함할 수 있다.

[0016] 일 실시예에 따른 펌웨어 퍼징 방법은 휴리스틱(heuristics) 기법에 기초하여 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 단계; 상기 시스템 모드 에뮬레이션 환경에서 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 단계; 상기 대상 프로세스 내 포함된 키워드에 기초하여 특정 형식을 갖춘 테스트 케이스를 생성하는 단계; 및 상기 테스트 케이스를 이용하여 상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 변이 기반 퍼징(fuzzing)을 실행하는 단계를 포함한다.

[0017] 상기 시스템 모드 에뮬레이션 환경을 제공하는 단계는, 상기 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스를 반복 실행하고, 반복 실행하여 획득된 결과 중 상기 시스템 모드 에뮬레이션 환경의 제공에 성공한 케이스에 기초하여 상기 시스템 모드 에뮬레이션 환경을 제공할 수 있다.

[0018] 상기 대상 프로세스는, 사용자로부터 입력을 요청하는 프로세스일 수 있다.

[0019] 상기 특정 형식은, 상기 대상 프로세스의 실행에 사용되는 데이터가 요구하는 형식일 수 있다.

[0020] 상기 특정 형식은, 상기 대상 프로세스에 입력되는 데이터가 요구하는 형식일 수 있다.

[0021] 일 실시예에 따른 펌웨어 퍼징 방법은 상기 퍼징을 실행하는 단계 이전에, 동적 기호 실행(concolic execution)을 통해 분기 조건을 해결하는 키워드를 상기 대상 프로세스에서 검출하는 단계를 더 포함할 수 있다.

## *발명의 효과*

[0022] 개시되는 실시예들에 따르면, 시스템 모드 에뮬레이터와 사용자 모드 에뮬레이터를 복합적으로 사용함으로써 호환성과 동시에 처리 속도가 향상시킬 수 있다.

[0023] 개시되는 실시예들에 따르면, 대상 프로세스에 사용되는 데이터가 요구하는 입력 형식에 기초하여 생성된 테스트 케이스를 통해 퍼징을 수행함으로써, 퍼징의 효율성 제고할 수 있다.

[0024] 개시되는 실시예들에 따르면, 퍼징 수행 시 테스팅부가 해결할 수 없는 인터럽트(interrupt)를 제2 테스팅부를 통해 해결함으로써 안정적인 퍼징을 수행할 수 있다.

## *도면의 간단한 설명*

[0025] 도 1은 일 실시예에 따른 펌웨어 퍼징 장치의 블록도

도 2는 추가적인 실시예에 따른 펌웨어 퍼징 장치를 설명하기 위한 흐름도

도 3은 일 실시예에 따른 펌웨어 퍼징 방법을 설명하기 위한 흐름도

도 4는 추가적인 실시예에 따른 펌웨어 퍼징 방법을 설명하기 위한 흐름도

도 5는 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

## *발명을 실시하기 위한 구체적인 내용*

[0026] 이하, 도면을 참조하여 일 실시예의 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.

[0027] 일 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 일 실시예의 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 일 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 성분들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 성분, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0028] 이하의 설명에 있어서, 신호 또는 정보의 "전송", "통신", "송신", "수신" 기타 이와 유사한 의미의 용어는 일 구성요소에서 다른 구성요소로 신호 또는 정보가 직접 전달되는 것뿐만이 아니라 다른 구성요소를 거쳐 전달되는 것도 포함한다.

[0029] 특히 신호 또는 정보를 일 구성요소로 "전송" 또는 "송신"한다는 것은 그 신호 또는 정보의 최종 목적지를 지시하는 것이고 직접적인 목적지를 의미하는 것이 아니다. 이는 신호 또는 정보의 "수신"에 있어서도 동일하다. 또한 본 명세서에 있어서, 2 이상의 데이터 또는 정보가 "관련"된다는 것은 하나의 데이터(또는 정보)를 획득하면, 그에 기초하여 다른 데이터(또는 정보)의 적어도 일부를 획득할 수 있음을 의미한다.

[0030] 또한, 제1, 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 목적으로 사용될 수 있다.

[0031] 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성 요소는 제2 구성 요소로 명명될 수 있고, 유사하게 제2 구성 요소도 제1 구성 요소로 명명될 수 있다.

[0032] 도 1은 일 실시예에 따른 펌웨어 퍼징 장치(100)의 블록도이다.

[0033] 도 1을 참조하면, 일 실시예에 따른 펌웨어 퍼징 장치(100)는 시스템 모드 에뮬레이터(110), 사용자 모드 에뮬레이터(120) 및 테스팅부(130)를 포함한다.

[0034] 이때, 시스템 모드 에뮬레이터(110), 사용자 모드 에뮬레이터(120) 및 테스팅부(130)은 각각 물리적으로 구분된 하나 이상의 장치를 이용하여 구현되거나, 하나 이상의 하드웨어 프로세서 또는 하나 이상의 하드웨어 프로세서 및 소프트웨어의 결합에 의해 구현될 수 있으며, 도시된 예와 달리 구체적 동작에 있어 명확히 구분되지 않을 수 있다.

[0035] 시스템 모드 에뮬레이터(110)는 휴리스틱(heuristics) 기법에 기초하여 사물 인터넷(IoT; Internet of Things)(이하, IoT) 기기에 시스템 모드 에뮬레이션 환경을 제공한다.

[0036] 여기서, 시스템 모드 에뮬레이션 환경이란, IoT 기기와 관련된 시스템 전체에 대한 에뮬레이팅(emulating)을 수행할 수 있는 환경을 의미한다.

[0037] 또한, 휴리스틱 기법이란, 경험에 기초하여 최적의 해답을 얻는 기법으로, 경험 기반의 귀납적 추론 기법에 해당한다.

[0038] 구체적으로, 시스템 모드 에뮬레이터(110)는 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스를 반복 실행하고, 반복 실행된 결과 중 시스템 모드 에뮬레이션 환경의 제공에 성공한 케이스에 기초하여 시스템 모드 에뮬레이션 환경을 제공함으로써, 휴리스틱 기법에 기초하여 시스템 모드 에뮬레이션 환경을 제공할 수 있다.

[0039] 이때, 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스는 예를 들어, 시스템 모드 에뮬레이터(110)가 펌웨어의 네트워크 서비스를 규정하지 못해 펌웨어에 대해 시스템 모드 에뮬레이션 환경의 제공에 실패하는 케이스를 포함할 수 있다.

[0040] 또 다른 예로, 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스는, 시스템 모드 에뮬레이터(110)가 주변 장치에 대해 시스템 모드 에뮬레이션 환경의 제공에 실패하는 케이스를 포함할 수 있다.

[0041] 또 다른 예로, 시스템 모드 에뮬레이션 환경의 제공 실패를 유발하는 하나 이상의 케이스는, 시스템 모드 에뮬레이터(110)가 지원할 수 없는 커널(kernel) 또는 부트 로더(boot loader)로 인해 시스템 모드 에뮬레이션 환경

의 제공을 실패하는 케이스를 포함할 수 있다.

[0042] 사용자 모드 에뮬레이터(120)는 시스템 모드 에뮬레이션 환경에서 IoT 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공한다.

[0043] 여기서, 사용자 모드 에뮬레이션 환경이란, 대상 프로세스에 대한 에뮬레이팅을 수행할 수 있는 환경을 의미한다.

[0044] 이때, 대상 프로세스란, 펌웨어를 실행함에 따라 발생되는 하나 이상의 프로세스 중 퍼징 수행의 타겟이 되는 프로세스를 의미한다.

[0045] 일 실시예에 따르면, 대상 프로세스는 사용자로부터 입력을 요청하는 프로세스일 수 있다. 예를 들어, 대상 프로세스는 웹 어플리케이션을 통해 사용자의 입력을 요청하는 프로세스일 수 있다.

[0046] 테스팅부(130)는 대상 프로세스 내 포함된 키워드에 기초하여 특정 형식을 갖춘 테스트 케이스를 생성하고, 테스트 케이스를 이용하여 사용자 모드 에뮬레이션 환경에서 대상 프로세스에 변이 기반 퍼징(fuzzing)을 실행한다.

[0047] 이하의 실시예에서 변이 기반 퍼징이란, 퍼징 수행 시 기 마련된 시드 파일(seed file)을 변형하여 테스트 케이스를 생성하는 퍼징 기법을 지칭한다.

[0048] 일 실시예에 따르면, 특정 형식이란, 대상 프로세스에 실행에 사용되는 데이터가 요구하는 형식일 수 있다. 구체적으로, 대상 프로세스에 실행에 사용되는 데이터는 대상 프로세스에 입력에 사용되는 데이터에 포함할 수 있다.

[0049] 예를 들어, 펌웨어가 웹 애플리케이션을 통해 사용자의 입력을 요청할 때 웹 애플리케이션의 입력에 사용되는 데이터가 특정 형식을 갖는 패킷을 구성하는 경우, 테스팅부(130)는 대상 프로세스로부터 해당 패킷 형식을 갖춘 테스트 케이스를 생성할 수 있다.

[0050] 도 2는 추가적인 실시예에 따른 펌웨어 퍼징 장치(100)를 설명하기 위한 흐름도이다.

[0051] 도 2를 참조하면, 추가적인 실시예에 따른 펌웨어 퍼징 장치(100)는 검출부(140)를 더 포함할 수 있다.

[0052] 다만, 도 2에 도시된 예에서 시스템 모드 에뮬레이터(110), 사용자 모드 에뮬레이터(120) 및 테스팅부(130)는 도 1에 도시된 구성과 동일한 구성이므로, 이에 대한 중복되는 설명은 생략하도록 한다.

[0053] 검출부(140)는 동적 기호 실행(concolic execution)을 통해 대상 프로세스에서 분기 조건을 해결하는 키워드를 검출한다.

[0054] 이하의 실시예들에서, 동적 기호 실행이란, 실제 수행(concrete execution)과 기호 실행(symbolic execution)을 모두 이용하여 경로 제약조건(path constraint)을 해결하는 기법으로, 동적 기호 실행은 동적 분석 및 정적 분석 기법을 결합하여 테스트 케이스를 자동으로 생성한다.

[0055] 구체적으로, 검출부(140)는 동적 기호 실행을 통해 테스트 케이스 생성 대상이 되는 프로세스와 초기의 테스트 케이스를 제공받아 초기 테스트 케이스를 실행한다. 이후, 실행된 프로그램의 경로를 분석하여 분기문에서 어떠한 조건이 수행되었는지를 나타내는 경로 분기 조건식을 생성한다.

[0056] 이때, 검출부(140)는 프로세스의 모든 실행 경로를 테스트하거나, 사용자가 지정한 종료 조건을 만족하면, 생성된 경로 분기 조건식을 분석함으로써 분기 조건을 해결하는 키워드를 추출할 수 있다.

[0057] 추가적인 실시예에 따르면, 펌웨어 퍼징 장치(100)는 제2 테스팅부(미도시)를 포함함으로써, 테스팅부(130)에서 실행되는 퍼징의 안정성을 보완시킬 수 있다.

[0058] 일 실시예에 따르면, 제2 테스팅부(미도시)는 퍼징의 실행 중 기 설정된 이벤트가 발생된 경우, 발생된 이벤트를 해결하기 위한 동작을 시스템 모드 에뮬레이션 환경에서 수행할 수 있다.

[0059] 일 실시예에 따르면, 제2 테스팅부(미도시)는 퍼징이 실행되는 도중 페이지 폴트(page fault)가 발생하는 경우, 페이지 폴트를 해결하기 위한 동작으로 페이지 폴트를 유발한 명령어를 실행하여 페이지 폴트를 처리할 수 있다.

[0060] 다른 실시예에 따르면, 제2 테스팅부(미도시)는 기 설정된 이벤트가 발생되는 경우로서, 퍼징이 실행되는 도중 하드웨어 의존성 함수의 실행이 요구되는 경우, 사전 생성된 라이브러리(library)를 이용하여 하드웨어 의존성

함수를 처리할 수 있다.

[0061] 여기서, 하드웨어 의존성 함수는 IoT 기기가 IoT 기기의 주변 기기로 접근을 가능하게 하는 함수를 의미한다.

[0062] 예를 들어, IoT 기기가 주변 기기인 NVRAM 장치로부터 config 파일을 읽는 함수를 nvram_get()라고 가정하면, 이때 하드웨어 의존 함수는 nvram_get()일 수 있다.

[0063] 다시 말해, 테스팅부(130)가 퍼징을 수행하는 도중 nvram_get() 함수의 실행이 요구되는 경우, 제2 테스팅부(미도시)는 라이브러리를 이용하여 nvram_get() 함수를 처리할 수 있다.

[0064] 이때, 라이브러리란, 하드웨어 의존성 함수의 결과를 참(true)으로 반환할 수 있도록 함수나 데이터들을 미리 만들어 모아 놓은 집합체이다.

[0065] 즉, 제2 테스팅부(미도시)는 nvram_get() 함수의 실행이 요구될 때 라이브러리를 통해 nvram_get() 함수의 결과값을 참으로 반환함에 따라 NVRAM 장치로 접근하지 않으면서도 nvram_get() 함수를 처리할 수 있다.

[0066] 이를 통해, 제2 테스팅부(미도시)는 주변 장치에 대해 에뮬레이션을 수행할 수 없는 에뮬레이터의 한계를 해결할 수 있다.

[0067] 이때, 테스팅부(130)는 기 설정된 이벤트가 발생한 경우, 퍼징의 실행을 중단하고, 동작의 수행 결과에 기초하여 퍼징의 재개 여부를 결정할 수 있다.

[0068] 구체적으로, 제2 테스팅부(미도시)의 동작의 수행 결과가 기 설정된 이벤트 해결에 성공에 해당하는 경우, 테스팅부(130)는 퍼징의 재개를 결정할 수 있다.

[0069] 도 3은 일 실시예에 따른 펌웨어 퍼징 방법을 설명하기 위한 흐름도이다.

[0070] 예시적으로, 도 3에 도시된 방법은 도 1에서 도시된 펌웨어 퍼징 장치(100)에 의해 수행될 수 있다.

[0071] 도 3을 참조하면, 펌웨어 퍼징 장치(100)는 휴리스틱(heuristics) 기법에 기초하여 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공한다(310).

[0072] 이후, 펌웨어 퍼징 장치(100)는 시스템 모드 에뮬레이션 환경에서 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공한다(320).

[0073] 이후, 펌웨어 퍼징 장치(100)는 대상 프로세스 내 포함된 키워드에 기초하여 특정 형식을 갖춘 테스트 케이스를 생성한다(330).

[0074] 이후, 펌웨어 퍼징 장치(100)는 테스트 케이스를 이용하여 사용자 모드 에뮬레이션 환경에서 대상 프로세스에 변이 기반 퍼징(fuzzing)을 실행한다(340).

[0075] 이후, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되었는지 여부를 판단한다(350).

[0076] 이때, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되지 않았다고 판단된 경우, 퍼징 종료 조건이 만족될 때까지 단계 330 및 340을 반복 수행한다.

[0077] 한편, 퍼징 종료 조건은 예를 들어, 퍼징이 실행된 횟수일 수 있으나 실시예에 따라 다양하게 설정될 수 있다.

[0078] 도 4는 추가적인 실시예에 따른 펌웨어 퍼징 방법을 설명하기 위한 흐름도이다.

[0079] 예시적으로, 도 4에 도시된 방법은 도 2에서 도시된 펌웨어 퍼징 장치(100)에 의해 수행될 수 있다.

[0080] 도 4를 참조하면, 펌웨어 퍼징 장치(100)는 휴리스틱 기법에 기초하여 사물 인터넷 기기에 시스템 모드 에뮬레이션 환경을 제공한다(410).

[0081] 이후, 펌웨어 퍼징 장치(100)는 시스템 모드 에뮬레이션 환경에서 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공한다(420).

[0082] 이후, 펌웨어 퍼징 장치(100)는 동적 기호 실행(concolic execution)을 통해 대상 프로세스에서 분기 조건을 해결하는 키워드를 검출한다(430).

[0083] 이후, 펌웨어 퍼징 장치(100)는 검출된 키워드에 기초하여 특정 형식을 갖춘 테스트 케이스를 생성한다(440).

[0084] 이후, 펌웨어 퍼징 장치(100)는 테스트 케이스를 이용하여 사용자 모드 에뮬레이션 환경에서 대상 프로세스에

변이 기반 퍼징(fuzzing)을 실행한다(450).

[0085] 이후, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되었는지 여부를 판단한다(460).

[0086] 이때, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되지 않았다고 판단된 경우, 퍼징 종료 조건이 만족될 때까지 단계 430 및 460을 반복 수행한다.

[0087] 한편, 퍼징 종료 조건은 예를 들어, 퍼징이 실행된 횟수일 수 있으나 실시예에 따라 다양하게 설정될 수 있다.

[0088] 또한, 도 3 및 도 4에 도시된 실시예는 복수 개의 단계로 나누어 기재되었으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다.

[0089] 도 5는 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경(10)을 예시하여 설명하기 위한 블록도이다.

[0090] 도 5는 일 실시예에 따르면 컴퓨팅 장치(12)를 포함하는 컴퓨팅 환경(10)을 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.

[0091] 도시된 컴퓨팅 환경(10)은 컴퓨팅 장치(12)를 포함한다. 일 실시예에서, 컴퓨팅 장치(12)는 펌웨어 퍼징 장치(100)에 포함된 하나 이상의 컴포넌트일 수 있다.

[0092] 컴퓨팅 장치(12)는 적어도 하나의 프로그램(14), 컴퓨터 판독 가능 저장 매체(16) 및 통신 버스(18)를 포함한다. 프로그램(14)는 컴퓨팅 장치(12)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로그램(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 프로그램(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0093] 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 프로그램(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.

[0094] 통신 버스(18)는 프로그램(14), 컴퓨터 판독 가능 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0095] 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.

[0096] 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 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리범위는 설명된 실시예에 국한되어 정해져서는 안 되며, 후술하는 청구범위뿐만 아니라 이 청구범위와 균등한 것들에 의해 정해져야 한다.

## *부호의 설명*

[0097] 10: 컴퓨팅 환경

12: 컴퓨팅 장치

14: 프로그램

16: 컴퓨터 판독 가능 저장 매체

18: 통신 버스

20: 프로그램

22: 입출력 인터페이스

24: 입출력 장치

26: 네트워크 통신 인터페이스

100: 펌웨어 퍼징 장치

110: 시스템 모드 에뮬레이터

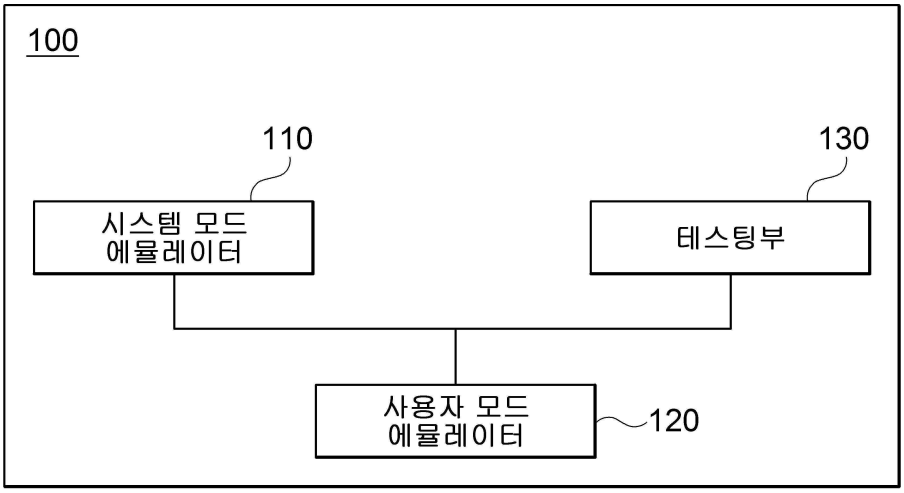
120: 사용자 모드 에뮬레이터

130: 테스팅부

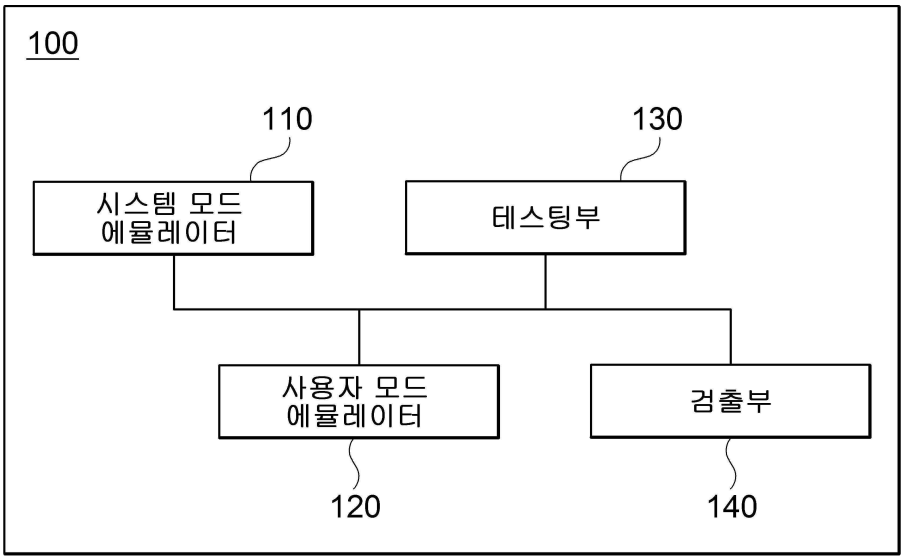
140: 검출부

## 도면

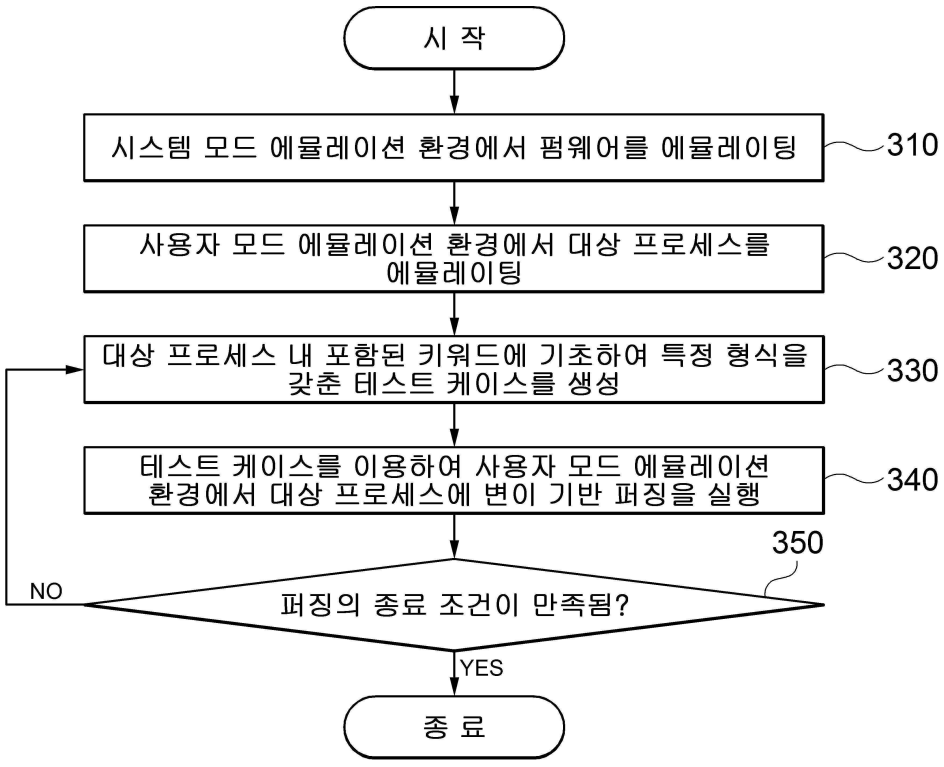
### 도면1



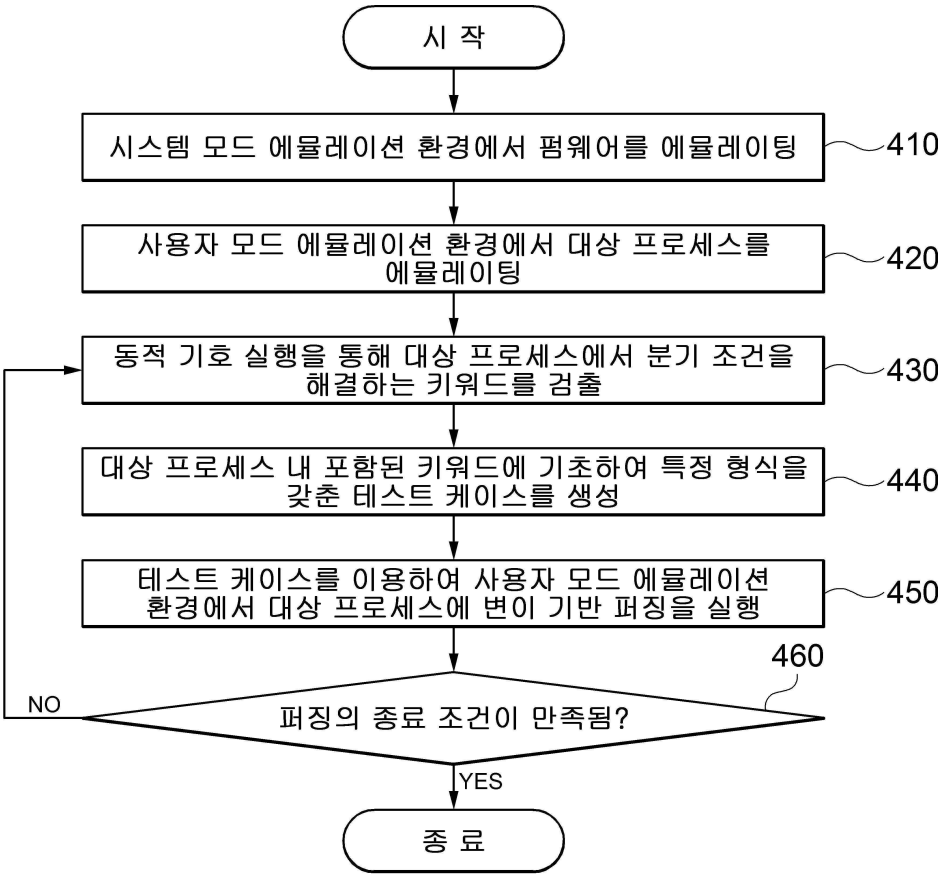
도면2


100
110
130
시스템 모드
에뮬레이터
테스팅부
사용자 모드
에뮬레이터
검출부
120
140

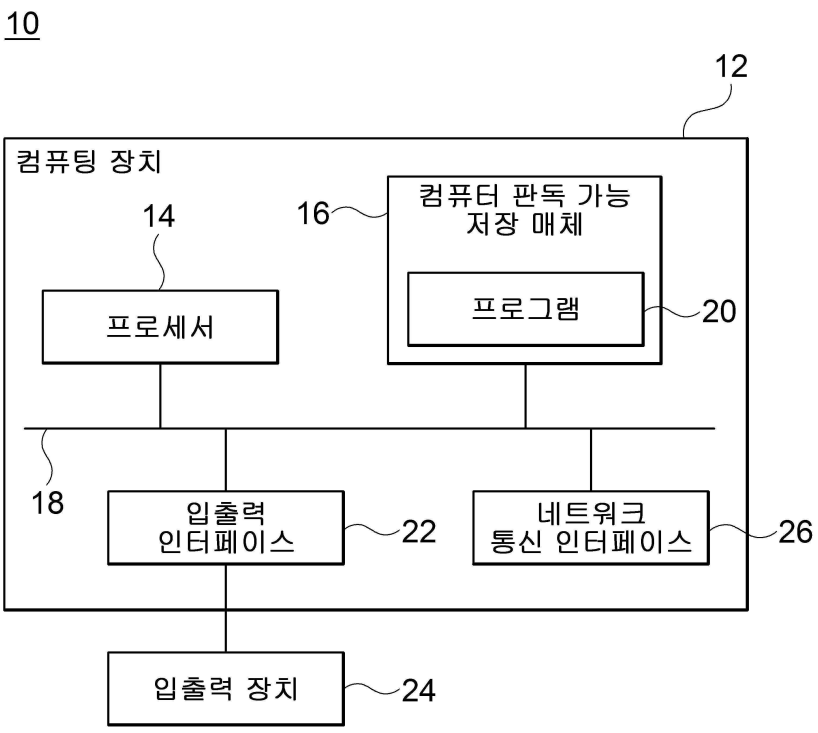
도면3


시 작
시스템 모드 에뮬레이션 환경에서 펌웨어를 에뮬레이팅
310
사용자 모드 에뮬레이션 환경에서 대상 프로세스를
에뮬레이팅
320
대상 프로세스 내 포함된 키워드에 기초하여 특정 형식을
갖춘 테스트 케이스를 생성
330
테스트 케이스를 이용하여 사용자 모드 에뮬레이션
환경에서 대상 프로세스에 변이 기반 퍼징을 실행
340
350
NO
퍼징의 종료 조건이 만족됨?
YES
종 료

도면4


시 작
시스템 모드 에뮬레이션 환경에서 펌웨어를 에뮬레이팅
410
사용자 모드 에뮬레이션 환경에서 대상 프로세스를 에뮬레이팅
420
동적 기호 실행을 통해 대상 프로세스에서 분기 조건을 해결하는 키워드를 검출
430
대상 프로세스 내 포함된 키워드에 기초하여 특정 형식을 갖춘 테스트 케이스를 생성
440
테스트 케이스를 이용하여 사용자 모드 에뮬레이션 환경에서 대상 프로세스에 변이 기반 퍼징을 실행
450
460
퍼징의 종료 조건이 만족됨?
NO
YES
종 료

도면5


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