(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년02월07일
(11) 등록번호 10-2635486
(24) 등록일자 2024년02월05일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) *G06F 11/36* (2006.01)
(52) CPC특허분류
G06F 21/577 (2013.01)
G06F 11/3684 (2013.01)
(21) 출원번호 10-2023-0000143
(22) 출원일자 2023년01월02일
심사청구일자 2023년01월02일
(56) 선행기술조사문헌
KR1020180086833 A*
US20220166699 A1*
'How to build a serial port fuzzer with Defensics SDK', Synopsys, Kari Hulkko, 2020.12.14.*
Synopsys, Kari Hulkko
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
이희조
박하람
(74) 대리인
이대호, 박건홍

전체 청구항 수 : 총 11 항 심사관 : 구대성

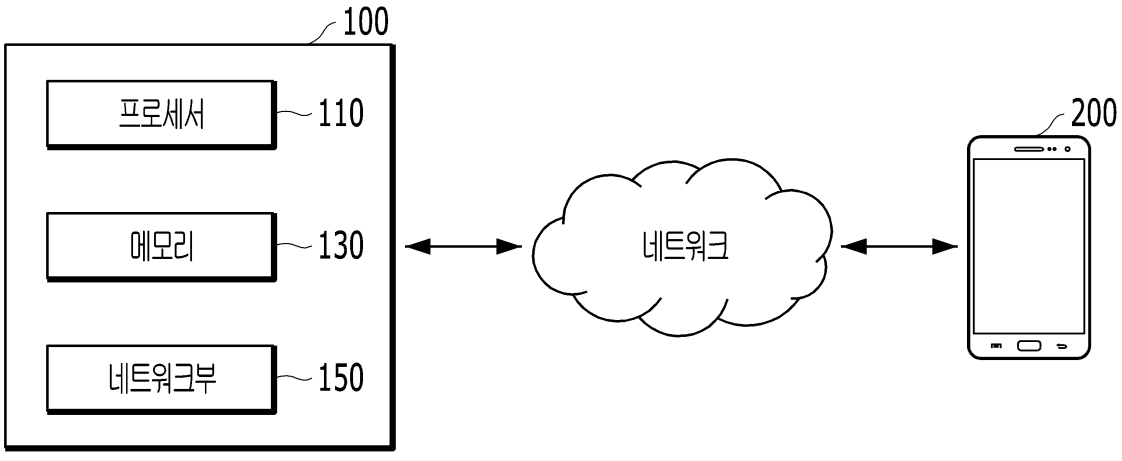
(54) 발명의 명칭 **퍼징을 수행하기 위한 방법 및 장치**

(57) 요 약

본 개시의 몇몇 실시예에 따라 프로세서를 포함하는 컴퓨팅 장치에서 퍼징(fuzzing)을 수행하기 위한 방법으로서, 디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택하는 단계; 상기 제 1 포트를 사전 결정된 제 1 상태(state)로 진입

(뒷면에 계속)

대 표 도 - 도1



시키기 위해, 상기 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계; 상기 제 1 포트로부터 수신한 상기 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 상기 제 1 포트의 제 1 현재 상태(state)를 결정하는 단계; 상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하는 단계; 및 상기 제 1 테스트 패킷을 이용하여 상기 제 1 포트의 퍼징 테스트를 수행하는 단계;를 포함할 수 있다.

(52) CPC특허분류
G06F 11/3688 (2013.01)
G06F 11/3692 (2013.01)
G06F 2221/033 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711160653
과제번호 2022-0-00277-001
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보보호핵심원천기술개발
연구과제명 SW공급망 보안을 위한 SBOM 자동생성 및 무결성 검증기술 개발
기 여 율 80/100
과제수행기관명 고려대학교산학협력단
연구기간 2023.01.01 ~ 2023.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711160660
과제번호 2022-0-01198-001
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 융합보안대학원(고려대학교)
기 여 율 10/100
과제수행기관명 고려대학교산학협력단
연구기간 2023.01.01 ~ 2023.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711193663
과제번호 2020-0-01819-004
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성
연구과제명 ICT명품인재양성(고려대학교)
기 여 율 10/100
과제수행기관명 고려대학교산학협력단
연구기간 2023.01.01 ~ 2023.12.31

공지예외적용 : 있음

명 세 서

청구범위

청구항 1

프로세서를 포함하는 컴퓨팅 장치에서 퍼징(fuzzing)을 수행하기 위한 방법으로서,

디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택하는 단계;

상기 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계;

상기 제 1 포트로부터 수신한 상기 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 상기 제 1 포트의 제 1 현재 상태(state)를 결정하는 단계;

상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하는 단계;

상기 제 1 테스트 패킷을 상기 제 1 포트에 전송하는 단계; 및

상기 제 1 포트로부터 수신한 상기 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 상기 제 1 포트의 취약점 발현 여부를 결정하는 단계;

를 포함하는,

방법.

청구항 2

제 1 항에 있어서,

상기 제 1 포트를 선택하는 단계는,

상기 적어도 하나의 포트 중에서 상기 페어링을 요구하고, 그리고 상기 페어링의 요구를 무시하고 통신이 가능한 상기 제 1 포트를 선택하는 단계;

를 포함하는,

방법.

청구항 3

제 1 항에 있어서,

상기 제 1 포트를 선택하는 단계는,

상기 적어도 하나의 포트 중에서 상기 페어링을 요구하지 않는 상기 제 1 포트를 선택하는 단계;

를 포함하는,

방법.

청구항 4

제 1 항에 있어서,

상기 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계 이전에,

작업(job)에 기초하여 상기 디바이스의 통신 프로토콜에 따라 사전 결정된 복수의 상태들을 복수의 그룹들로 분류하는 단계; 및

상기 복수의 그룹들 각각에 대응되는 명령어를 매핑시키는 단계;

를 포함하는,

방법.

청구항 5

제 1 항에 있어서,

상기 명령어에 포함된 적어도 하나의 필드는,

고정되어 변경 불가능한 데이터를 포함하는 고정 필드(fixed field);

다른 필드에 종속된 데이터를 포함하는 종속 필드(dependent field); 및

변형 가능한 데이터를 포함하는 변형 필드(mutable field);

를 포함하는,

방법.

청구항 6

제 5 항에 있어서,

상기 변형 필드는,

상기 디바이스의 통신 채널 또는 포트 중 적어도 하나와 관련된 데이터를 포함하는 핵심 필드(core field); 및

응용 소프트웨어와 관련된 데이터를 포함하는 응용 필드(application field);

를 포함하는,

방법.

청구항 7

제 6 항에 있어서,

상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 상기 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하는 단계는,

상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 명령어에 포함된 핵심 필드를 변형시킨 상기 제 1 테스트 패킷을 생성하는 단계;

를 포함하는,

방법.

청구항 8

삭제

청구항 9

제 1 항에 있어서,

상기 제 1 포트로부터 수신한 상기 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 상기 제 1 포트의 취약점 발현 여부를 결정하는 단계는,

상기 제 2 응답 패킷과 사전 저장된 정상 응답 패킷이 서로 상이한 경우, 상기 제 1 포트에 상기 제 1 테스트 패킷에 의한 취약점이 발현된 것으로 결정하는 단계; 및

발현된 상기 취약점에 관한 정보를 생성하는 단계;

를 포함하는,

방법.

청구항 10

제 1 항에 있어서,

상기 제 1 포트를 상기 제 1 상태와 상이한 사전 결정된 제 2 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 2 상태에 대응되는 명령어가 매핑(mapping)된 제 2 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계;

상기 제 1 포트로부터 수신한 상기 제 2 상태 전이 패킷에 대한 제 3 응답 패킷에 기초하여 상기 제 1 포트의 제 2 현재 상태(state)를 결정하는 단계;

상기 제 1 포트의 제 2 현재 상태가 상기 사전 결정된 제 2 상태에 대응되는 경우, 상기 사전 결정된 제 2 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 2 테스트 패킷을 생성하는 단계; 및

상기 제 2 테스트 패킷을 이용하여 상기 제 1 포트의 퍼징 테스트를 수행하는 단계;

를 더 포함하는,

방법.

청구항 11

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은 퍼징(fuzzing)을 수행하기 위한 컴퓨팅 장치의 프로세서로 하여금 이하의 단계들을 수행하기 위한 명령들을 포함하며, 상기 단계들은:

디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택하는 단계;

상기 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계;

상기 제 1 포트로부터 수신한 상기 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 상기 제 1 포트의 제 1 현재 상태(state)를 결정하는 단계;

상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하는 단계;

상기 제 1 테스트 패킷을 상기 제 1 포트에 전송하는 단계; 및

상기 제 1 포트로부터 수신한 상기 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 상기 제 1 포트의 취약점 발현 여부를 결정하는 단계;

를 포함하는,

컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램.

청구항 12

퍼징(fuzzing)을 수행하기 위한 컴퓨팅 장치에 있어서,

적어도 하나의 코어를 포함하는 프로세서;

상기 프로세서에 의해 실행가능한 컴퓨터 프로그램을 저장하는 메모리; 및

네트워크부;

를 포함하고,

상기 프로세서는,

디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택하고,

상기 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하고,

상기 제 1 포트로부터 수신한 상기 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 상기 제 1 포트의 제 1 현재 상태(state)를 결정하고,

상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하고,

상기 제 1 테스트 패킷을 상기 제 1 포트에 전송하고, 그리고

상기 제 1 포트로부터 수신한 상기 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 상기 제 1 포트의 취약점 발현 여부를 결정하는,

컴퓨팅 장치.

발명의 설명

기 술 분 야

[0001] 본 개시내용은 퍼징 수행하기 위한 방법 및 장치에 관한 것으로, 보다 구체적으로 디바이스의 포트에 대한 퍼징을 수행하기 위한 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 블루투스(Bluetooth)란 디지털 통신 기기(예를 들어, 스마트 폰, 태블릿 PC 등)를 위한 개인 근거리 무선 통신 산업 표준이다. 블루투스는 ISM 대역에 포함되는 2.4GHz를 이용하며 최대 100m 거리를 둔 기기 간의 정보를 교환하는데 사용될 수 있다.

[0003] 블루투스는 Bluetooth Special Interest Group(SIG)를 통해 관리되며 블루투스 기기로 인증을 받기 위해서는 SIG에서 제정한 블루투스 표준 규격에 만족해야 한다. 전기통신, 컴퓨터, 네트워크, 가전 관련 기기를 제조하는 기업들은 SIG에서 공개한 블루투스 문서를 참조해 블루투스 프로토콜 스택을 구현할 수 있다.

[0004] 블루투스를 사용하는 장치는 블루투스 프로토콜 스택의 취약점을 효과적으로 탐지하기 위해 퍼징(fuzzing)을 활용할 수 있다.

[0005] 퍼징이란 소프트웨어에서 알지 못하지만 발생할 수 있는 취약점을 찾기 위해 무작위의 값을 주입하여 비정상적 행동을 출력하는지 검증하는 기법일 수 있다. 따라서, 블루투스를 사용하는 장치는 블루투스 프로토콜 스택에서 발생될 수 있는 문제점을 획득하기 위해 퍼징 테스트를 사용할 수 있다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 대한민국 등록특허 10-2190727(2020.12.08. 등록)

발명의 내용

해결하려는 과제

[0007] 본 개시는 전술한 배경기술에 대응하여 안출된 것으로, 디바이스의 포트에 대한 퍼징을 수행하기 위함이다.

[0008] 본 개시의 기술적 과제들은 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0009] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 몇몇 실시예에 따라, 프로세서를 포함하는 컴퓨팅 장치에서 퍼징(fuzzing)을 수행하기 위한 방법으로서, 디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택하는 단계; 상기 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계; 상기 제 1 포트로부터 수신한 상기 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 상기 제 1 포트의 제 1 현재 상태(state)를 결정하는 단계; 상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하는 단계; 및 상기 제 1 테스트 패킷을 이용하여 상기 제 1 포트의 퍼징 테스트를 수행하는 단계;를 포함할 수 있다.

[0010] 대안적으로, 상기 제 1 포트를 선택하는 단계는, 상기 적어도 하나의 포트 중에서 상기 페어링을 요구하고, 그리고 상기 페어링의 요구를 무시하고 통신이 가능한 상기 제 1 포트를 선택하는 단계;를 포함할 수 있다.

[0011] 대안적으로, 상기 제 1 포트를 선택하는 단계는, 상기 적어도 하나의 포트 중에서 상기 페어링을 요구하지 않는 상기 제 1 포트를 선택하는 단계;를 포함할 수 있다.

[0012] 대안적으로, 상기 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계 이전에, 작업(job)에 기초하여 상기 디바이스의 통신 프로토콜에 따라 사전 결정된 복수의 상태들을 복수의 그룹들로 분류하는 단계; 및 상기 복수의 그룹들 각각에 대응되는 명령어를 매핑시키는 단계;를 포함할 수 있다.

[0013] 대안적으로, 상기 명령어에 포함된 적어도 하나의 필드는, 고정되어 변경 불가능한 데이터를 포함하는 고정 필드(fixed field); 다른 필드에 종속된 데이터를 포함하는 종속 필드(dependent field); 및 변형 가능한 데이터를 포함하는 변형 필드(mutable field);를 포함할 수 있다.

[0014] 대안적으로, 상기 변형 필드는, 상기 디바이스의 통신 채널 또는 포트 중 적어도 하나와 관련된 데이터를 포함하는 핵심 필드(core field); 및 응용 소프트웨어와 관련된 데이터를 포함하는 응용 필드(application field);를 포함할 수 있다.

[0015] 대안적으로, 상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 상기 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하는 단계는, 상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 명령어에 포함된 핵심 필드를 변형시킨 상기 제 1 테스트 패킷을 생성하는 단계;를 포함할 수 있다.

[0016] 대안적으로, 상기 제 1 테스트 패킷을 이용하여 상기 제 1 포트의 퍼징 테스트를 수행하는 단계는, 상기 제 1 테스트 패킷을 상기 제 1 포트에 전송하는 단계; 및 상기 제 1 포트로부터 수신한 상기 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 상기 제 1 포트의 취약점 발현 여부를 결정하는 단계;를 포함할 수 있다.

[0017] 대안적으로, 상기 제 1 포트로부터 수신한 상기 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 상기 제 1 포트의 취약점 발현 여부를 결정하는 단계는, 상기 제 2 응답 패킷과 사전 저장된 정상 응답 패킷이 서로 상이한 경우, 상기 제 1 포트에 상기 제 1 테스트 패킷에 의한 취약점이 발현된 것으로 결정하는 단계; 및 발현된 상기 취약점에 관한 정보를 생성하는 단계;를 포함할 수 있다.

[0018] 대안적으로, 상기 제 1 포트를 상기 제 1 상태와 상이한 사전 결정된 제 2 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 2 상태에 대응되는 명령어가 매핑(mapping)된 제 2 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계; 상기 제 1 포트로부터 수신한 상기 제 2 상태 전이 패킷에 대한 제 3 응답 패킷에 기초하여 상기 제 1 포트의 제 2 현재 상태(state)를 결정하는 단계; 상기 제 1 포트의 제 2 현재 상태가 상기 사전 결정된 제 2 상태에 대응되는 경우, 상기 사전 결정된 제 2 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 2 테스트 패킷을 생성하는 단계; 및 상기 제 2 테스트 패킷을 이용하여 상기 제 1 포트의 퍼징 테스트를 수행하는 단계;를 더 포함할 수 있다.

[0019] 대안적으로, 컴퓨터 판독가능 저장 매체에 저장된 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램은 퍼징(fuzzing)을 수행하기 위한 컴퓨팅 장치의 프로세서로 하여금 이하의 단계들을 수행하기 위한 명령들을 포함하며, 상기 단계들은: 디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택하는 단계; 상기 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하는 단계; 상기 제 1 포트로부터 수신한 상기 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 상기 제 1 포트의 제 1 현재 상태(state)를 결정하는 단계; 상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하는 단계; 및 상기 제 1 테스트 패킷을 이용하여 상기 제 1 포트의 퍼징 테스트를 수행하는 단계;를 포함할 수 있다.

[0020] 대안적으로, 퍼징(fuzzing)을 수행하기 위한 컴퓨팅 장치에 있어서, 적어도 하나의 코어를 포함하는 프로세서; 상기 프로세서에 의해 실행가능한 컴퓨터 프로그램을 저장하는 메모리; 및 네트워크부;를 포함하고, 상기 프로세서는, 디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택하고, 상기 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 상기 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 상기 제 1 포트에 전송하고, 상기 제 1 포트로부터 수신한 상기 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 상기 제 1 포트의 제 1 현재 상태(state)를 결정하고, 상기 제 1 포트의 제 1 현재 상태가 상기 사전 결정된 제 1 상태에 대응되는 경우, 상기 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성하고, 그리고 상기 제 1 테스트 패킷을 이용하여 상기 제 1 포트의 퍼징 테스트를 수행할 수 있다.

발명의 효과

[0021] 본 개시의 일 실시예에 따른 기법은 디바이스의 포트에 대한 퍼징을 수행할 수 있다.

[0022] 본 개시에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0023] 다양한 양상들이 이제 도면들을 참조로 기재되며, 여기서 유사한 참조 번호들은 총괄적으로 유사한 구성요소들을 지칭하는데 이용된다. 이하의 실시예에서, 설명 목적을 위해, 다수의 특정 세부사항들이 하나 이상의 양상들의 총체적 이해를 제공하기 위해 제시된다. 그러나, 그러한 양상(들)이 이러한 구체적인 세부사항들 없이 실시될 수 있음은 명백할 것이다.

도 1은 본 개시의 몇몇 실시예에 따른 퍼징을 수행하기 위한 시스템을 개략적으로 나타낸 도면이다.

도 2는 본 개시의 몇몇 실시예에 따른 명령어의 구조를 개략적으로 나타낸 도면이다.

도 3은 본 개시의 몇몇 실시예에 따른 퍼징을 수행하기 위한 컴퓨팅 장치에 의해 테스트 패킷을 생성하는 과정을 나타낸 도면이다.

도 4는 본 개시의 몇몇 실시예에 따른 퍼징을 수행하기 위한 컴퓨팅 장치의 프로그램과 종래의 기술들의 상태 커버리지(state coverage)를 나타낸 도면이다.

도 5는 본 개시의 몇몇 실시예에 따라 퍼징을 수행하기 위한 방법을 예시적으로 나타낸 도면이다.

도 6은 본 개시의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0024] 다양한 실시예들이 이제 도면을 참조하여 설명된다. 본 명세서에서, 다양한 설명들이 본 개시의 이해를 제공하기 위해서 제시된다. 그러나, 이러한 실시예들은 이러한 구체적인 설명 없이도 실행될 수 있음이 명백하다.

[0025] 본 명세서에서 사용되는 용어 "컴포넌트", "모듈", "시스템" 등은 컴퓨터-관련 엔티티, 하드웨어, 펌웨어, 소프트웨어, 소프트웨어 및 하드웨어의 조합, 또는 소프트웨어의 실행을 지칭한다. 예를 들어, 컴포넌트는 프로세서 상에서 실행되는 처리과정(procedure), 프로세서, 객체, 실행 스레드, 프로그램, 및/또는 컴퓨터일 수 있지만, 이들로 제한되는 것은 아니다. 예를 들어, 컴퓨팅 장치에서 실행되는 애플리케이션 및 컴퓨팅 장치 모두 컴포넌트일 수 있다. 하나 이상의 컴포넌트는 프로세서 및/또는 실행 스레드 내에 상주할 수 있다. 일 컴포넌트는 하나의 컴퓨터 내에 로컬화 될 수 있다. 일 컴포넌트는 2개 이상의 컴퓨터들 사이에 분배될 수 있다. 또한, 이러한 컴포넌트들은 그 내부에 저장된 다양한 데이터 구조들을 갖는 다양한 컴퓨터 판독가능한 매체로부터 실행할 수 있다. 컴포넌트들은 예를 들어 하나 이상의 데이터 패킷들을 갖는 신호(예를 들면, 로컬 시스템, 분산 시스템에서 다른 컴포넌트와 상호작용하는 하나의 컴포넌트로부터의 데이터 및/또는 신호를 통해 다른 시스템과 인터넷과 같은 네트워크를 통해 전송되는 데이터)에 따라 로컬 및/또는 원격 처리들을 통해 통신할 수 있다.

[0026] 더불어, 용어 "또는"은 배타적 "또는"이 아니라 내포적 "또는"을 의미하는 것으로 의도된다. 즉, 달리 특정되지 않거나 문맥상 명확하지 않은 경우에, "X는 A 또는 B를 이용한다"는 자연적인 내포적 치환 중 하나를 의미하는 것으로 의도된다. 즉, X가 A를 이용하거나; X가 B를 이용하거나; 또는 X가 A 및 B 모두를 이용하는 경우, "X는 A 또는 B를 이용한다"가 이들 경우들 어느 것으로도 적용될 수 있다. 또한, 본 명세서에 사용된 "및/또는"이라는 용어는 열거된 관련 아이템들 중 하나 이상의 아이템의 가능한 모든 조합을 지칭하고 포함하는 것으로 이해되어야 한다.

[0027] 또한, "포함한다" 및/또는 "포함하는"이라는 용어는, 해당 특징 및/또는 구성요소가 존재함을 의미하는 것으로 이해되어야 한다. 다만, "포함한다" 및/또는 "포함하는"이라는 용어는, 하나 이상의 다른 특징, 구성요소 및/또는 이들의 그룹의 존재 또는 추가를 배제하지 않는 것으로 이해되어야 한다. 또한, 달리 특정되지 않거나 단수 형태를 지시하는 것으로 문맥상 명확하지 않은 경우에, 본 명세서와 청구범위에서 단수는 일반적으로 "하나 또는 그 이상"을 의미하는 것으로 해석되어야 한다.

[0028] 그리고, "A 또는 B 중 적어도 하나"이라는 용어는, "A만을 포함하는 경우", "B 만을 포함하는 경우", "A와 B의 구성으로 조합된 경우"를 의미하는 것으로 해석되어야 한다.

[0029] 당업자들은 추가적으로 여기서 개시된 실시예들과 관련되어 설명된 다양한 예시적 논리적 블록들, 구성들, 모듈들, 회로들, 수단들, 로직들, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양쪽 모두의 조합들로 구현될 수 있음을 인식해야 한다. 하드웨어 및 소프트웨어의 상호교환성을 명백하게 예시하기 위해, 다양한 예시적 컴포넌트들, 블록들, 구성들, 수단들, 로직들, 모듈들, 회로들, 및 단계들은 그들의 기능성 측면에서 일반적으로 위에서 설명되었다. 그러한 기능성이 하드웨어로 또는 소프트웨어로서 구현되는지 여부는 전반적인 시스템에 부과된 특정 어플리케이션(application) 및 설계 제한들에 달려 있다. 숙련된 기술자들은 각각의 특정 어플리케이션들을 위해 다양한 방법들로 설명된 기능성을 구현할 수 있다. 다만, 그러한 구현의 결정들이 본 개시내용의 영역을 벗어나게 하는 것으로 해석되어서는 안된다.

[0030] 제시된 실시예들에 대한 설명은 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이다. 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 발명은 여기에 제시된 실시예 들로 한정되는 것이 아니다. 본 발명은 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

[0031] 제시된 실시예들에 대한 설명은 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이다. 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 발명은 여기에 제시된 실시예 들로 한정되는 것이 아니다. 본 발명은 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

[0032] 본 개시내용에서의 제 1, 제 2, 또는 제 3 과 같이 제 N 으로 표현되는 용어들은 복수의 엔티티들을 구분하기 위해 사용된다. 예를 들어, 제 1 과 제 2로 표현된 엔티티들은 서로 동일하거나 또는 상이할 수 있다. 제 1-1, 제 1-2로 표현되는 용어들 그리고 제 2-1, 제 2-2로 표현되는 용어들 또한 복수의 엔티티들을 서로 구분하기 위해 사용될 수 있다.

[0033] 본 개시내용에서 퍼징(fuzzing) 및/또는 퍼징 테스트는 소프트웨어에서 알지 못하지만 발생할 수 있는 취약점을 찾기 위해 무작위의 값 및/또는 사전 결정된 공격 패턴을 주입하여 비정상적 행동을 출력하는지 검증하는 기법을 의미할 수 있다.

[0034] 본 개시내용에서 통신 프로토콜은 블루투스 프로토콜 스택을 포함할 수 있다. 예를 들어, 통신 프로토콜은 블루투스 L2CAP 프로토콜을 포함할 수 있다.

[0035] 본 개시내용에서의 패킷은 통신 프로토콜에서 사용되는 포맷(format)의 데이터를 포함할 수 있다. 예를 들어, 패킷은 블루투스 L2CAP 프로토콜에서 사용되는 포맷의 데이터를 포함할 수 있다.

[0037] 도 1은 본 개시의 몇몇 실시예에 따른 퍼징(fuzzing)을 수행하기 위한 시스템을 개략적으로 나타낸 도면이다.

[0038] 도 1을 참조하면, 퍼징을 수행하기 위한 시스템은 컴퓨팅 장치(100), 디바이스(200), 네트워크 등을 포함할 수 있다.

[0039] 도 1에 도시된 시스템의 구성은 간략화 하여 나타낸 예시일 뿐이다. 본 개시의 일 실시예에서 시스템은 퍼징을 수행하기 위한 다른 구성들이 포함될 수 있고, 개시된 구성들 중 일부만이 시스템을 구성할 수도 있다.

[0040] 컴퓨팅 장치(100)는 프로세서(110), 메모리(130), 네트워크부(150)를 포함할 수 있다. 예를 들어, 컴퓨팅 장치(100)는 서버, 사용자 단말 등을 포함할 수 있다.

[0041] 프로세서(110)는 하나 이상의 코어로 구성될 수 있으며, 컴퓨팅 장치의 중앙 처리 장치(CPU: central processing unit), 범용 그래픽 처리 장치 (GPGPU: general purpose graphics processing unit), 텐서 처리 장치(TPU: tensor processing unit) 등의 데이터 분석 및 처리를 위한 프로세서를 포함할 수 있다. 프로세서(110)는 메모리(130)에 저장된 컴퓨터 프로그램을 판독하여 본 개시의 일 실시예에 따른 데이터 분석 및 처리를 수행할 수 있다.

[0042] 프로세서(110)는 통상적으로 컴퓨팅 장치(100)의 전반적인 동작을 제어할 수 있다. 프로세서(110)는 컴퓨팅 장치(100)에 포함된 구성요소들을 통해 입력 또는 출력되는 신호, 데이터, 정보 등을 처리하거나 메모리(130)에 저장된 응용 프로그램을 구동함으로써, 사용자에게 적절한 정보 및/또는 기능을 제공 및/또는 처리할 수 있다.

[0043] 프로세서(110)는 메모리(130)에 저장된 응용 프로그램을 구동하기 위하여, 컴퓨팅 장치(100)의 구성요소들 중 적어도 일부를 제어할 수 있다. 나아가, 프로세서(110)는 응용 프로그램의 구동을 위하여, 컴퓨팅 장치(100)에 포함된 구성요소들 중 적어도 둘 이상을 서로 조합하여 동작시킬 수 있다.

[0044] 본 개시의 일 실시예에 따르면, 메모리(130)는 프로세서(110)가 생성하거나 결정한 임의의 형태의 정보 및/또는 네트워크부(150)가 수신한 임의의 형태의 정보를 저장할 수 있다.

[0045] 본 개시의 일 실시예에 따르면, 메모리(130)는 플래시 메모리 타입(flash memory type), 하드디스크 타입(hard disk type), 멀티미디어 카드 마이크로 타입(multimedia card micro type), 카드 타입의 메모리(예를 들어 SD 또는 XD 메모리 등), 램(Random Access Memory, RAM), SRAM(Static Random Access Memory), 롬(Read-Only Memory, ROM), EEPROM(Electrically Erasable Programmable Read-Only Memory), PROM(Programmable Read-Only Memory), 자기 메모리, 자기 디스크 및/또는 광디스크 중 적어도 하나의 타입의 저장매체를 포함할 수 있다. 컴퓨팅 장치(100)는 인터넷(internet) 상에서 메모리(130)의 저장 기능을 수행하는 웹 스토리지(web storage)와 관련되어 동작할 수도 있다. 전술한 메모리에 대한 기재는 예시일 뿐, 본 개시는 이에 제한되지 않는다.

[0046] 본 개시의 일 실시예에 따른 네트워크부(150)는 임의의 형태의 데이터 및 신호 등을 송수신할 수 있는 임의의 유무선 통신 네트워크가 본 개시 내용에서 표현되는 네트워크에 포함될 수 있다. 본 명세서에서 설명된 기술들은 위에서 언급된 네트워크들뿐만 아니라, 다른 네트워크들에서도 사용될 수 있다.

[0048] 프로세서(110)는 디바이스(200)에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 적어도 하나의 포트 중에서 제 1 포트를 선택할 수 있다.

[0049] 적어도 하나의 포트는 통신 프로토콜에서 애플리케이션이 상호 통신을 위해 사용하는 가상의 논리적 통신 연결

단일 수 있다. 예를 들어, 적어도 하나의 포트는 디바이스(200)에서 제공하는 서비스 포트를 포함할 수 있다.

[0050] 페어링은 무선 연결하기 위해 연결을 원하는 기기의 식별정보(예를 들어, 연결을 원하는 기기의 하드웨어 정보, 네트워크 정보, 사용자 정보 등)를 등록하는 것을 의미할 수 있다. 페어링의 요구는 장치 또는 장치에 포함된 포트와 연결을 원하는 기기 간에 서로 식별정보의 등록을 요구하는 것을 의미할 수 있다. 예를 들어, 디바이스(200)(및/또는 디바이스(200)의 포트)는 컴퓨팅 장치(100)와 무선 연결을 위해 컴퓨팅 장치(100)에 페어링을 요구할 수 있다. 컴퓨팅 장치(100)가 페어링을 요구하는 디바이스(200)(및/또는 디바이스(200)의 포트)와 페어링을 하는 경우, 디바이스(200)(및/또는 디바이스(200)의 포트)와 컴퓨팅 장치(100)는 각각 식별정보를 등록하여 서로 무선 연결할 수 있다.

[0051] 본 개시의 몇몇 실시예에 따르면, 프로세서(110)는 적어도 하나의 포트 중에서 페어링을 요구하고, 그리고 페어링의 요구를 무시하고 통신이 가능한 제 1 포트를 선택할 수 있다. 프로세서(110)는 페어링을 요구하는 포트들 중에서 페어링의 요구를 무시하고 통신이 가능한 포트를 제 1 포트로 결정할 수 있다. 페어링을 요구하는 포트들 중에서 페어링의 요구를 무시하고 통신이 가능한 포트가 복수인 경우, 프로세서(110)는 사전 결정된 기준(예를 들어, 포트에 할당된 포트 번호의 오름차순(또는 내림차순), 탐색된 시간 순서 등)에 기초하여 페어링을 요구하는 포트들 중에서 페어링의 요구를 무시하고 통신이 가능한 제 1 포트를 결정할 수 있다.

[0052] 본 개시의 몇몇 다른 실시예에 따르면, 프로세서(110)는 적어도 하나의 포트 중에서 페어링을 요구하지 않는 제 1 포트를 선택할 수 있다. 페어링을 요구하지 않는 포트가 복수인 경우, 프로세서(110)는 사전 결정된 기준(예를 들어, 포트에 할당된 포트 번호의 오름차순(또는 내림차순), 탐색된 시간 순서 등)에 기초하여 페어링을 요구하지 않는 제 1 포트를 결정할 수 있다.

[0053] 프로세서(110)가 페어링을 요구하는 포트로 연결을 하는 경우, 포트를 통해 기기(예를 들어, 디바이스(200))를 장악할 수 있는 권한을 얻을 수 있다. 하지만, 일반적으로 포트에 대한 공격은 권한이 없는 상태에서 수행될 수 있다. 따라서, 이미 권한이 있는 상태에서 퍼징을 수행하는 것은 유효한 취약점을 발견하기에 적절하지 않을 수 있다. 그러므로, 프로세서(110)는 페어링을 요구하지 않는 포트 또는 페어링의 요구가 있더라도 무시하고 통신이 가능한 포트로 연결하여 퍼징을 수행함으로써, 포트의 유효한 취약점을 획득할 수 있다.

[0055] 프로세서(110)는 작업(job)에 기초하여 디바이스(200)의 통신 프로토콜에 따라 사전 결정된 복수의 상태들을 복수의 그룹들로 분류할 수 있다.

[0056] 작업(job)은 컴퓨터에서 처리해야할 일과 관련된 프로그램의 집합일 수 있다. 작업은 이벤트(event), 기능(function), 액션(action)을 포함할 수 있다. 이벤트는 수신 장치(예를 들어, 디바이스(200))에서 송신 장치(예를 들어, 컴퓨팅 장치(100))로부터 받은 패킷에 의해 발생되는 일과 관련된 프로그램의 집합일 수 있다. 기능은 이벤트를 처리하는 일과 관련된 프로그램의 집합일 수 있다. 액션은 패킷에 대한 응답을 수신 장치(예를 들어, 디바이스(200))에서 송신 장치(예를 들어, 컴퓨팅 장치(100))로 보내는 일과 관련된 프로그램의 집합일 수 있다. 예를 들어, 작업은 특정 일을 종료하기 위한 Closed, 특정 일을 연결하기 위한 Connection, 특정 일을 생성하기 위한 Creation, 특정 일을 배열하기 위한 Configuration, 특정 일을 중단하기 위한 Disconnection, 특정 일을 이동시키기 위한 Move, 특정 일을 시작하기 위한 Open 등을 포함할 수 있다. 다만, 작업의 종류는 이에 한정되지 않는다.

[0057] 사전 결정된 복수의 상태들은 시스템이나 프로그램에 어떤 작업을 수행시키기 위해 설정되는 조건이나 방법일 수 있다. 따라서, 디바이스(200)의 통신 프로토콜에서 작동하는 방법에 따라서 상태가 변화될 수 있다. 복수의 그룹들은 아래의 표 1과 같이 분류될 수 있다.

표 1

[0058]

Job	States
Closed	{CLOSED}
Connection	{WAIT CONNECT, WAIT CONNECT RSP}
Creation	{WAIT CREATE, WAIT CREATE RSP}
Configuration	{WAIT CONFIG, WAIT CONFIG RSP, WAIT CONFIG REQ, WAIT CONFIG REQ RSP, WAIT SEND CONFIG, WAIT IND FINAL RSP, WAIT FINAL RSP, WAIT CONTROL IND}
Disconnection	{WAIT DISCONNECT}
Move	{WAIT MOVE, WAIT MOVE RSP, WAIT MOVE CONFIRM, WAIT CONFIRM RSP}
Open	{OPEN}

[0059] 표 1을 참조하면, 작업 Closed에 대응되는 상태는 {CLOSED}로 하나의 그룹일 수 있다. 작업 Connection에 대응되는 상태는 {WAIT CONNECT, WAIT CONNECT RSP}로 하나의 그룹일 수 있다. 작업 Creation에 대응되는 상태는 {WAIT CREATE, WAIT CREATE RSP}으로 하나의 그룹일 수 있다. 작업 Configuration에 대응되는 상태는 {WAIT CONFIG, WAIT CONFIG RSP, WAIT CONFIG REQ, WAIT CONFIG REQ RSP, WAIT SEND CONFIG, WAIT IND FINAL RSP, WAIT FINAL RSP, WAIT CONTROL IND}로 하나의 그룹일 수 있다. 작업 Disconnection에 대응되는 상태는 {WAIT DISCONNECT}로 하나의 그룹일 수 있다. 작업 Move에 대응되는 상태는 {WAIT MOVE, WAIT MOVE RSP, WAIT MOVE CONFIRM, WAIT CONFIRM RSP}로 하나의 그룹일 수 있다. 작업 Open에 대응되는 상태는 {OPEN}으로 하나의 그룹일 수 있다.

[0061] 프로세서(110)는 복수의 그룹들 각각에 대응되는 명령어를 매핑시킬 수 있다. 복수의 그룹들 각각에 대응되는 명령어는 아래의 표 2와 같이 매핑될 수 있다.

표 2

[0062]

Job	Valid commands
Closed	All commands
Connection	Connect Req/Rsp
Creation	Create Channel Req/Rsp
Configuration	Config Req/Rsp
Disconnection	Disconnect Req/Rsp
Move	Move Channel Req/Rsp,Move Channel Confirmation Req/Rsp
Open	All commands

[0063] 표 2를 참조하면, 작업 Closed에 대응되는 명령어는 모든 명령어(All commands)일 수 있다. 작업 Connection에 대응되는 명령어는 Connect Req 또는 Rsp일 수 있다. 작업 Creation에 대응되는 명령어는 Create Channel Req 또는 Rsp일 수 있다. 작업 Configuration에 대응되는 명령어는 Config Req 또는 Rsp일 수 있다. 작업 Disconnection에 대응되는 명령어는 Disconnect Req 또는 Rsp일 수 있다. 작업 Move에 대응되는 명령어는 Move Channel Req 또는 Rsp, Move Channel Confirmation Req 또는 Rsp일 수 있다. 작업 Open에 대응되는 명령어는 모든 명령어(All commands)일 수 있다.

[0065] 프로세서(110)는 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 제 1 포트에 전송할 수 있다.

[0066] 예를 들어, 사전 결정된 제 1 상태가 {WAIT DISCONNECT}인 경우, 프로세서(110)는 {WAIT DISCONNECT}에 대응되는 명령어인 Disconnect Req 또는 Rsp가 매핑된 제 1 상태 전이 패킷을 제 1 포트에 전송할 수 있다.

[0067] 상태 전이 패킷은 통신 프로토콜의 상태를 변화시키는 패킷일 수 있다. 예를 들어, 제 1 상태 전이 패킷은 제 1 포트의 상태를 현재 {CLOSED}에서 {WAIT DISCONNECT}로 변화시키는 패킷일 수 있다.

[0068] 제 1 포트는 제 1 상태 전이 패킷을 수신할 수 있다. 제 1 포트는 제 1 상태 전이 패킷에 의해 제 1 상태로 상태 전이될 수 있다.

[0070] 프로세서(110)는 제 1 포트로부터 수신한 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 제 1 포트의 제 1 현재 상태(state)를 결정할 수 있다.

[0071] 응답 패킷은 수신한 패킷에 대한 응답으로 생성된 패킷으로, 수신한 디바이스의 현재 상태에 대한 정보를 포함할 수 있다. 예를 들어, 제 1 응답 패킷은 디바이스(200)의 제 1 포트에서 제 1 상태 전이 패킷에 대한 응답으로 생성된 패킷일 수 있다. 제 1 응답 패킷은 디바이스(200)의 제 1 포트의 제 1 현재 상태에 대한 정보를 포함할 수 있다. 따라서, 프로세서(110)는 제 1 응답 패킷에 포함된 제 1 포트의 제 1 현재 상태에 대한 정보에 기초하여 제 1 포트의 제 1 현재 상태를 결정할 수 있다.

[0073] 프로세서(110)는 제 1 포트의 제 1 현재 상태가 사전 결정된 제 1 상태에 대응되는 경우, 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성할 수 있다.

[0074] 명령어는 프로그래밍 언어에서 적어도 하나의 동작을 수행하는 데이터일 수 있다. 명령어는 적어도 하나의 동작을 수행하기 위한 적어도 하나의 필드를 포함할 수 있다.

[0075] 적어도 하나의 필드는 고정 필드(fixed field), 종속 필드(dependent field), 변형 필드(mutable field) 등을

포함할 수 있다.

[0076] 고정 필드(fixed field)는 변경 불가능한 데이터를 포함할 수 있다. 예를 들어, 고정 필드는 HEADER CID 필드 등을 포함할 수 있다.

[0077] 종속 필드(dependent field)는 다른 필드에 종속된 데이터를 포함할 수 있다. 종속 필드에 포함되는 데이터는 다른 필드의 데이터에 기초하여 결정될 수 있다. 예를 들어, 종속 필드는 PAYLOAD LEN 필드, CODE 필드, ID 필드, DATA LEN 필드 등을 포함할 수 있다.

[0078] 변형 필드(mutable field)는 변형 가능한 데이터를 포함할 수 있다. 변형 필드에 포함되는 데이터는 사전 저장된 필드 참조 데이터에 기초하여 결정될 수 있다. 변형 필드에 포함되는 데이터는 사용자에 의해 결정될 수 있다. 예를 들어, 변형 필드는 핵심 필드(core field), 응용 필드(application field) 등을 포함할 수 있다.

[0079] 핵심 필드는 디바이스(200)의 통신 채널 또는 포트 중 적어도 하나와 관련된 데이터를 포함할 수 있다. 예를 들어, 핵심 필드는 PSM 필드, SCID 필드, DCID 필드, ICID 필드, CONT ID 필드 등을 포함할 수 있다.

[0080] 응용 필드는 응용 소프트웨어와 관련된 데이터를 포함할 수 있다. 예를 들어, 응용 필드는 REASON 필드, RESULT 필드, STATUS 필드, FLAGS 필드, TYPE 필드, INTERVAL 필드, LATENCY 필드, TIMEOUT 필드, SPSM 필드, MTU 필드, CREDIT 필드, MPS 필드, OPT 필드, QoS 필드 등을 포함할 수 있다.

[0081] 테스트 패킷은 퍼징을 위해 명령어에 포함된 적어도 하나의 필드를 변형시킨 패킷일 수 있다. 예를 들어, 제 1 테스트 패킷은 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨 패킷일 수 있다.

[0082] 테스트 패킷은 명령어에 포함된 필드 중에서 프로토콜의 핵심 기능을 담당하는 핵심 필드를 변형시킨 패킷일 수 있다. 예를 들어, 제 1 테스트 패킷은 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 필드 중에서 프로토콜의 핵심 기능을 담당하는 핵심 필드를 변형시킨 패킷일 수 있다.

[0083] 테스트 패킷(예를 들어, 제 1 테스트 패킷)은 사전 저장된 필드 참조 데이터에 기초하여 핵심 필드를 변형시킨 패킷일 수 있다. 사전 저장된 필드 참조 데이터는 통신 프로토콜에 포함된 필드 각각의 가용 데이터를 포함할 수 있다.

[0084] 테스트 패킷(예를 들어, 제 1 테스트 패킷)의 핵심 필드에 변형된 데이터가 삽입되고, 그리고 테스트 패킷(예를 들어, 제 1 테스트 패킷)의 응용 필드에 기본 값(default value)이 삽입될 수 있다. 따라서, 프로세서(110)는 테스트 패킷(예를 들어, 제 1 테스트 패킷)의 핵심 필드에 변형된 데이터를 삽입시키고, 그리고 테스트 패킷의 응용 필드에 기본 값을 삽입시킬 수 있다. 본 개시의 몇몇 실시예에 따르면, 프로세서(110)는 테스트 패킷(예를 들어, 제 1 테스트 패킷)에 쓰레기 값(garbage value)을 추가로 삽입시킬 수 있다. 테스트 패킷(예를 들어, 제 1 테스트 패킷)에 쓰레기 값을 추가로 삽입함으로써, 대상 포트에서 취약점 발현 확률이 높아질 수도 있다.

[0085] 본 개시의 몇몇 실시예에 따르면, 프로세서(110)는 제 1 포트의 제 1 현재 상태가 사전 결정된 제 1 상태에 대응되는 경우, 명령어에 포함된 핵심 필드를 변형시킨 제 1 테스트 패킷을 생성할 수 있다. 본 개시의 몇몇 다른 실시예에 따르면, 프로세서(110)는 명령어에 포함된 핵심 필드가 복수인 경우, 명령어에 포함된 복수의 핵심 필드들 중에서 적어도 하나를 변형시킨 제 1 테스트 패킷을 생성할 수 있다. 따라서, 프로세서(110)는 핵심 필드를 변형시킨 테스트 패킷(예를 들어, 제 1 테스트 패킷)을 생성함으로써, 퍼징 테스트의 대상이 되는 포트에서 거부(reject)되지 않는 테스트 패킷을 생성할 수 있다.

[0087] 프로세서(110)는 제 1 테스트 패킷을 이용하여 제 1 포트의 퍼징 테스트를 수행할 수 있다.

[0088] 예를 들어, 프로세서(110)는 제 1 테스트 패킷을 제 1 포트에 전송할 수 있다. 제 1 테스트 패킷은 제 1 상태에 대응되는 명령어에서 핵심 필드만 변형시킨 패킷이므로, 제 1 포트에서 거부되지 않을 수 있다.

[0089] 프로세서(110)는 제 1 포트로부터 수신한 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 제 1 포트의 취약점 발현 여부를 결정할 수 있다.

[0090] 일례로, 프로세서(110)는 제 2 응답 패킷과 사전 저장된 정상 응답 패킷이 서로 상이한 경우, 제 1 포트에 제 1 테스트 패킷에 의한 취약점이 발현된 것으로 결정할 수 있다. 사전 저장된 정상 응답 패킷은 포트(예를 들어, 제 1 포트, 제 2 포트 등)별로 정상 상태인 경우에 수신되는 응답 패킷을 포함할 수 있다. 사전 저장된 정상 응답 패킷은 포트별로 사전에 정상 패킷을 전송하여 수신된 응답 패킷을 포함할 수 있다. 사전 저장된 정상 응답 패킷은 통신 프로토콜에서 통신을 하며 수집된 패킷을 포함할 수 있다. 사전 저장된 정상 응답 패킷은 포트별로

사전에 정상으로 결정되어 있는 패킷을 포함할 수 있다.

[0091] 다른 일례로, 프로세서(110)는 제 2 응답 패킷에 에러 메시지 및/또는 크래시 덤프(crash dump)가 포함된 경우, 제 1 포트에 제 1 테스트 패킷에 의한 취약점이 발현된 것으로 결정할 수 있다. 에러 메시지는 사전 저장된 정상 응답 패킷에 포함된 정상 메시지와 상이한 내용을 포함하는 메시지일 수 있다. 예를 들어, 에러 메시지는 “연결이 끊겼습니다.”, “응답이 없습니다.” 등의 내용을 포함할 수 있다. 크래시 덤프는 특정 프로그램이 비정상적으로 종료된 시점과 관련된 상태 정보를 포함할 수 있다. 예를 들어, 크래시 덤프는 특정 프로그램이 비정상적으로 종료된 시점에서의 프로그램 정보, 프로세서 정보, 메모리 정보 등을 포함할 수 있다. 크래시 덤프는 특정 프로그램이 비정상적으로 종료되는 경우에 생성될 수 있다. 따라서, 프로세서(110)는 크래시 덤프를 분석하여 프로그램의 취약점을 획득할 수 있다.

[0092] 프로세서(110)는 제 1 테스트 패킷에 의해 발현된 취약점에 관한 정보를 생성할 수 있다. 제 1 테스트 패킷에 의해 발현된 취약점에 관한 정보는 취약점, 취약점 발현 원인, 취약점을 발생시킨 테스트 패킷, 취약점이 발생된 포트, 취약점이 발현된 시간 등을 포함할 수 있다.

[0093] 프로세서(110)는 생성된 취약점에 관한 정보를 메모리(130)에 저장할 수 있다. 본 개시의 몇몇 실시예에 따르면, 프로세서(110)는 생성된 취약점에 관한 정보를 디바이스(200)에 전송할 수 있다. 따라서, 디바이스(200)의 사용자는 취약점에 관한 정보를 확인하고 대처 방안을 강구할 수 있다.

[0095] 프로세서(110)는 제 1 포트를 제 1 상태와 상이한 사전 결정된 제 2 상태(state)로 진입시키기 위해, 사전 결정된 제 2 상태에 대응되는 명령어가 매핑(mapping)된 제 2 상태 전이 패킷을 제 1 포트에 전송할 수 있다.

[0096] 예를 들어, 사전 결정된 제 2 상태가 {WAIT CONNECT}인 경우, 프로세서(110)는 {WAIT CONNECT}에 대응되는 명령어인 Connect Req 또는 Rsp가 매핑된 제 2 상태 전이 패킷을 제 1 포트에 전송할 수 있다.

[0097] 제 2 상태 전이 패킷은 제 2 포트의 상태를 현재 {CLOSED}에서 {WAIT CONNECT}로 변화시키는 패킷일 수 있다.

[0098] 제 1 포트는 제 2 상태 전이 패킷을 수신할 수 있다. 제 1 포트는 제 2 상태 전이 패킷에 의해 제 2 상태로 상태 전이될 수 있다.

[0100] 프로세서(110)는 제 1 포트로부터 수신한 제 2 상태 전이 패킷에 대한 제 3 응답 패킷에 기초하여 제 1 포트의 제 2 현재 상태(state)를 결정할 수 있다.

[0101] 제 3 응답 패킷은 디바이스(200)의 제 1 포트에서 제 2 상태 전이 패킷에 대한 응답으로 생성된 패킷일 수 있다. 제 3 응답 패킷은 디바이스(200)의 제 1 포트의 제 2 상태에 대한 정보를 포함할 수 있다. 따라서, 프로세서(110)는 제 3 응답 패킷에 포함된 제 2 현재 상태에 대한 정보에 기초하여 제 1 포트의 제 2 현재 상태를 결정할 수 있다.

[0103] 프로세서(110)는 제 1 포트의 제 2 현재 상태가 사전 결정된 제 2 상태에 대응되는 경우, 사전 결정된 제 2 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 2 테스트 패킷을 생성할 수 있다.

[0104] 제 2 테스트 패킷은 사전 결정된 제 2 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨 패킷일 수 있다. 제 2 테스트 패킷은 사전 결정된 제 2 상태에 대응되는 명령어에 포함된 필드 중에서 프로토콜의 핵심 기능을 담당하는 핵심 필드를 변형시킨 패킷일 수 있다.

[0105] 제 2 테스트 패킷은 사전 저장된 필드 참조 데이터에 기초하여 핵심 필드를 변형시킨 패킷일 수 있다. 사전 저장된 필드 참조 데이터는 통신 프로토콜에 제시된 필드별 사용 데이터를 포함할 수 있다.

[0106] 제 2 테스트 패킷의 핵심 필드에 변형된 데이터가 삽입되고, 그리고 제 2 테스트 패킷의 응용 필드에 기본 값(default value)이 삽입될 수 있다. 따라서, 프로세서(110)는 제 2 테스트 패킷의 핵심 필드에 변형된 데이터를 삽입시키고, 그리고 제 2 테스트 패킷의 응용 필드에 기본 값을 삽입시킬 수 있다. 본 개시의 몇몇 실시예에 따르면, 프로세서(110)는 제 2 테스트 패킷에 쓰레기 값(garbage value)을 추가로 삽입시킬 수 있다. 제 2 테스트 패킷에 쓰레기 값을 추가로 삽입함으로써, 대상 포트에서 취약점 발현 확률이 높아질 수도 있다.

[0107] 본 개시의 몇몇 실시예에 따르면, 프로세서(110)는 제 1 포트의 제 2 현재 상태가 사전 결정된 제 2 상태에 대응되는 경우, 명령어에 포함된 핵심 필드를 변형시킨 제 2 테스트 패킷을 생성할 수 있다. 본 개시의 몇몇 다른 실시예에 따르면, 프로세서(110)는 명령어에 포함된 핵심 필드가 복수인 경우, 명령어에 포함된 복수의 핵심 필드들 중에서 적어도 하나를 변형시킨 제 2 테스트 패킷을 생성할 수 있다. 따라서, 프로세서(110)는 핵심 필드를 변형시킨 테스트 패킷을 생성함으로써, 퍼징 테스트의 대상이 되는 포트에서 거부(reject)되지 않는 테스트

패킷을 생성할 수 있다.

[0109] 프로세서(110)는 제 2 테스트 패킷을 이용하여 제 1 포트의 퍼징 테스트를 수행할 수 있다.

[0110] 예를 들어, 프로세서(110)는 제 2 테스트 패킷을 제 1 포트에 전송할 수 있다. 제 2 테스트 패킷은 제 1 상태에 대응되는 명령어에서 핵심 필드만 변형시킨 패킷이므로, 제 1 포트에서 거부되지 않을 수 있다.

[0111] 프로세서(110)는 제 1 포트로부터 수신한 제 2 테스트 패킷에 대한 제 4 응답 패킷에 기초하여 제 1 포트의 취약점 발현 여부를 결정할 수 있다.

[0112] 일례로, 프로세서(110)는 제 4 응답 패킷과 사전 저장된 정상 응답 패킷이 서로 상이한 경우, 제 1 포트에 제 2 테스트 패킷에 의한 취약점이 발현된 것으로 결정할 수 있다. 사전 저장된 정상 응답 패킷은 포트(예를 들어, 제 1 포트, 제 2 포트 등)별로 정상 상태인 경우에 수신되는 응답 패킷을 포함할 수 있다.

[0113] 프로세서(110)는 제 2 테스트 패킷에 의해 발현된 취약점에 관한 정보를 생성할 수 있다. 제 2 테스트 패킷에 의해 발현된 취약점에 관한 정보는 취약점, 취약점 발현 원인, 취약점을 발생시킨 테스트 패킷, 취약점이 발생된 포트, 취약점이 발현된 시간 등을 포함할 수 있다.

[0114] 프로세서(110)는 생성된 취약점에 관한 정보를 메모리(130)에 저장할 수 있다. 본 개시의 몇몇 실시예에 따르면, 프로세서(110)는 생성된 취약점에 관한 정보를 디바이스(200)에 전송할 수 있다. 따라서, 디바이스(200)의 사용자는 취약점에 관한 정보를 확인하고 대처 방안을 강구할 수 있다.

[0116] 프로세서(110)는 디바이스(200)에 포함된 적어도 하나의 포트 각각의 페어링의 요구 여부에 기초하여 적어도 하나의 포트 중에서 제 1 포트와 상이한 제 2 포트를 선택할 수 있다.

[0117] 프로세서(110)는 제 2 포트를 사전 결정된 제 3 상태(state)로 진입시키기 위해, 사전 결정된 제 3 상태에 대응되는 명령어가 매핑(mapping)된 제 3 상태 전이 패킷을 제 2 포트에 전송할 수 있다.

[0118] 프로세서(110)는 제 2 포트로부터 수신한 제 3 상태 전이 패킷에 대한 제 5 응답 패킷에 기초하여 제 2 포트의 제 3 현재 상태(state)를 결정할 수 있다.

[0119] 프로세서(110)는 제 2 포트의 제 3 현재 상태가 사전 결정된 제 1 상태에 대응되는 경우, 사전 결정된 제 3 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 3 테스트 패킷을 생성할 수 있다.

[0120] 프로세서(110)는 제 3 테스트 패킷을 이용하여 제 2 포트의 퍼징 테스트를 수행할 수 있다.

[0121] 프로세서(110)는 상술한 바와 같이 디바이스(200)에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 선택된 포트들 각각에 대해서 퍼징 테스트를 수행할 수 있다.

[0123] 디바이스(200)는 컴퓨팅 장치(100)와 통신을 위한 매커니즘을 갖는 시스템에서의 임의의 형태의 장치를 의미할 수 있다. 예를 들어, 디바이스(200)는 컴퓨팅 장치(100)와 네트워크를 통해 패킷을 송신 및/또는 수신할 수 있다. 본 개시의 몇몇 실시예에 따라, 컴퓨팅 장치(100)에 디바이스(200)가 포함되는 경우, 디바이스(200)는 컴퓨팅 장치(100)와 네트워크를 사용 또는 비사용하여 내부에서 패킷을 송신 및/또는 수신할 수 있다. 디바이스(200)는 컴퓨팅 장치(100)로부터 수신한 패킷에 기초하여 정보를 처리할 수 있다.

[0124] 디바이스(200)는 통신 프로토콜로 블루투스 프로토콜 스택을 포함할 수 있다. 예를 들어, 디바이스(200)는 블루투스 L2CAP 프로토콜을 포함할 수 있다. 컴퓨팅 장치(100) 및/또는 디바이스(200)는 각각 블루투스 통신을 위한 장치를 포함할 수 있다. 예를 들어, 컴퓨팅 장치(100) 및/또는 디바이스(200)는 각각 블루투스 통신을 위한 장치(예를 들어, 동글(dongle) 등), 회로 등을 포함하거나 또는 별도로 구비하고 있을 수 있다.

[0126] 네트워크는 컴퓨팅 장치(100) 및 디바이스(200)가 서로 임의의 형태의 데이터 및 신호를 송수신할 수 있는 임의의 유무선 통신 네트워크를 포함할 수 있다. 예를 들어, 컴퓨팅 장치(100)는 네트워크를 통해 유선 및/또는 무선으로 디바이스(200)에 패킷을 송신할 수 있다. 다른 예를 들어, 컴퓨팅 장치(100)는 네트워크를 통해 유선 및/또는 무선으로 디바이스(200)로부터 패킷을 수신할 수 있다.

[0128] 도 2는 본 개시의 몇몇 실시예에 따른 명령어의 구조를 개략적으로 나타낸 도면이다.

[0129] 도 2를 참조하면, 명령어는 PAYLOAD LEN 필드(10), HEADER CID 필드(20), CODE 필드(30), ID 필드(40), DATA LEN 필드(50), DATA 필드(60) 등을 포함할 수 있다.

[0130] PAYLOAD LEN 필드(10)는 종속 필드에 포함되고, 정보 페이로드(CODE 필드(30), ID 필드(40), DATA LEN 필드

(50) 및 DATA 필드(60))의 길이에 기초하여 결정된 데이터를 포함할 수 있다. PAYLOAD LEN 필드(10)는 정보 페이로드의 길이에 관한 데이터를 포함할 수 있다.

[0131] HEADER CID 필드(20)는 고정 필드에 포함되고, 사전 결정된 특정 데이터를 포함할 수 있다. 예를 들어, 사전 결정된 특정 데이터는 ACL-U 논리 링크(logical links)에서 사용되는 "0x0001"일 수 있다.

[0132] CODE 필드(30)는 종속 필드에 포함되고, 유효한 명령 코드(valid command code)에 의해 결정된 데이터를 포함할 수 있다.

[0133] ID 필드(40)는 종속 필드에 포함되고, 장치(예를 들어, 컴퓨팅 장치(100), 디바이스(200) 등)에 의해 동적으로 할당된 데이터를 포함할 수 있다.

[0134] DATA LEN 필드(50) 종속 필드에 포함되고, DATA 필드(60)에 포함된 데이터의 길이에 의해 결정된 데이터를 포함할 수 있다.

[0135] DATA 필드(60)는 변형 필드에 포함되고, 핵심 필드(61) 및 응용 필드(62)를 포함할 수 있다.

[0136] 핵심 필드(61)는 PSM 필드, SCID 필드, DCID 필드, ICID 필드, CONT ID 필드 등을 포함할 수 있다. PSM 필드는 포트 설정에 사용될 수 있다. SCID 필드, DCID 필드, ICID 필드 및 CONT ID 필드는 채널 종점(channel endpoint)의 설정에 사용될 수 있다.

[0137] 응용 필드(62)는 REASON 필드, RESULT 필드, STATUS 필드, FLAGS 필드, TYPE 필드, INTERVAL 필드, LATENCY 필드, TIMEOUT 필드, SPSM 필드, MTU 필드, CREDIT 필드, MPS 필드, OPT 필드, QoS 필드 등을 포함할 수 있다.

[0138] 응용 필드(62)에 포함된 각각의 필드들은 명령에 대한 응용 프로그램 데이터가 포함될 수 있다. 응용 필드(62)에 포함된 각각의 필드들은 포트 또는 채널 관리에 영향을 주지 않고 데이터를 전달하기 위한 필드일 수 있다.

[0140] 도 3은 본 개시의 몇몇 실시예에 따른 퍼징을 수행하기 위한 컴퓨팅 장치에 의해 테스트 패킷을 생성하는 과정을 나타낸 도면이다.

[0141] 도 3을 참조하면, 정상 패킷은 PAYLOAD LEN 필드(10a), HEADER CID 필드(20a), CODE 필드(30a), ID 필드(40a), DATA LEN 필드(50a), 핵심 필드(예를 들어, DCID 필드 등)(61a), 응용 필드(예를 들어, MTU 필드 등)(62a)를 포함할 수 있다.

[0142] 프로세서(110)는 테스트 패킷(예를 들어, 제 1 테스트 패킷, 제 2 테스트 패킷 등)을 생성하기 위해 고정 필드 또는 종속 필드인 PAYLOAD LEN 필드(10b), HEADER CID 필드(20b), CODE 필드(30b), ID 필드(40b) 및 DATA LEN 필드(50b)에 정상 패킷에 대응되는 값을 각각 삽입할 수 있다.

[0143] 프로세서(110)는 테스트 패킷(예를 들어, 제 1 테스트 패킷, 제 2 테스트 패킷 등)을 생성하기 위해 핵심 필드(예를 들어, DCID 필드 등)(61b)에 사전 저장된 필드 참조 데이터에 기초하여 정상 패킷의 핵심 필드(61a)와 상이한 변형 데이터인 "8F7B"를 삽입할 수 있다.

[0144] 프로세서(110)는 테스트 패킷(예를 들어, 제 1 테스트 패킷, 제 2 테스트 패킷 등)을 생성하기 위해 응용 필드(예를 들어, MTU 필드 등)(62b)에 기본 값(default value)인 "00 00 00 00 00 00"을 삽입할 수 있다.

[0145] 프로세서(110)는 테스트 패킷(예를 들어, 제 1 테스트 패킷, 제 2 테스트 패킷 등)을 생성하기 위해 무작위(random) 또는 사전 결정된 값으로 결정된 쓰레기 값(garbage value)(70)을 삽입할 수 있다.

[0147] [실험예 1]

[0148] 본 개시의 몇몇 실시예에 따라 퍼징을 수행하기 위한 컴퓨팅 장치(100)에 의해 생성 및 저장된 프로그램(이하, L2Fuzz)은 외부 라이브러리를 제외하고 약 1200줄의 Python 코드로 구현될 수 있다. 예를 들어, L2Fuzz는 패킷 변형을 위한 대화형 패킷 조작 프로그램인 Scapy 라이브러리(v2.4.4)를 사용하여 구현될 수 있다. L2Fuzz는 Ubuntu 18.04.4 LTS, 8GB 메모리, Intel Core i5-7500 CPU @ 3.40GHz × 4, 50GB 디스크 및 Billionton Bluetooth Class 1 동글이 있는 가상 머신에서 실행될 수 있다.

[0149] 실험 대상 장치는 BlueZ(Linux), BlueDroid(Android), Apple BT 스택 및 Windows BT 스택을 포함하여 범용 Bluetooth 프로토콜 스택을 나타낼 수 있는 8개의 실제 테스트 장치를 선택할 수 있다. 8개의 실제 테스트 장치는 아래의 표 3과 같을 수 있다.

표 3

[0150]

No.	Type	Vendor	Name	Year	Model	Chip	OS or FW	BT Stack	BT Ver.
D1	Tablet PC	Google	Nexus 7	2013	ASUS-1A005A	Snapdragon 600	Android 6.0.1	BlueDroid	4.0 + LE
D2	Smartphone	Google	Pixel 3	2018	GA00464	Snapdragon 845	Android 11.0.1	BlueDroid	5.0 + LE
D3	Smartphone	Samsung	Galaxy 7	2016	SM-G930L	Exynos 8890	Android 8.0.0	BlueDroid	4.2
D4	Smartphone	Apple	iPhone 6S	2015	A1688	A9	iOS 15.0.2	iOS stack	4.2
D5	Earphone	Apple	Airpods 1 gen	2016	A1523	W1	6.8.8	RTKit stack	4.2
D6	Earphone	Samsung	Galaxy Buds+	2020	SM-R175NZKATUR	BCM43015	R175XXU0AUG1	BTW	5.0 + LE
D7	Laptop	LG	Gram	2019	15ZD990-VX50K	Intel wireless BT	Windows 10	Windows stack	5.0
D8	Laptop	LG	Gram	2017	15ZD970-GX55K	Intel wireless BT	Ubuntu 18.04.4	BlueZ	5.0

[0151] L2Fuzz의 효과를 평가할 때 L2Fuzz의 결과는 종래 기술인 Defensics, BFuzz 및 BSS의 결과와 비교했다. 다른 관련 기술은 L2CAP 취약점 탐지를 지원하지 않거나 공개적으로 사용할 수 없기 때문에 제외되었다. 테스트 장치 D2(표 3 참조)를 사용하여 L2Fuzz의 변형 효율성 및 상태 적용 범위를 기본 퍼저(fuzzer)(예를 들어, Defensics, BFuzz 및 BSS)와 비교할 수 있다. D2는 커스텀이 거의 없는 블루투스 표준을 따르는 디바이스일 수 있다. 따라서, D2는 평가에 사용될 수 있다.

[0152] BlueZ와 BlueDroid를 제외한 대부분의 블루투스 스택은 폐쇄형 소스이므로 블루투스 퍼저는 블랙박스 퍼저에 가깝다. 따라서, 소스 코드 커버리지와 같은 화이트박스 또는 그레이박스 퍼징에 사용되는 평가 메트릭스(evaluation metrics)는 사용하기 어렵다. 그러므로, 블루투스 퍼저를 평가하기 위한 평가 지표는 패킷 추적만으로 측정할 수 있는 변형 효율성(mutation efficiency) 및 상태 커버리지(state coverage)를 사용할 수 있다.

[0153] 변형 효율성은 거부 없이 전송된 malformed packet(예를 들어, 테스트 패킷)의 최소 비율일 수 있다. 변형 효율성은 아래의 수학식 1을 통해 산출될 수 있다.

수학식 1

[0154]
$$Mutation\ efficiency = MP\ Ratio * (1 - PR\ Ratio)$$

[0155] MP Ratio는 malformed packet의 비율일 수 있다. PR Ratio는 패킷 거부 비율일 수 있다. MP Ratio는 아래의 수학식 2를 통해 산출될 수 있다.

수학식 2

[0156]
$$MP\ Ratio = \frac{\#Transmitted\ Malformed\ Packets}{\#Transmitted\ Packets}$$

[0157] MP Ratio는 전송된 malformed packet의 개수를 전송된 전체 패킷의 개수로 나눈 값일 수 있다.

[0158] PR Ratio는 아래의 수학식 3을 통해 산출될 수 있다.

수학식 3

[0159]
$$PR\ Ratio = \frac{\#Received\ Rejection\ Packets\ from\ Target}{\#Received\ Packets\ from\ Target}$$

[0160] PR Ratio는 타겟(예를 들어, 디바이스(200)의 제 1 포트, 제 2 포트 등)으로부터 수신된 거절 패킷의 개수를 타

겟으로부터 수신된 패킷의 개수로 나눈 값일 수 있다.

[0161] 상태 커버리지는 커버가 되는 L2CAP 상태의 수를 의미할 수 있다. 취약점은 상태 전이 과정과 각 상태의 기능(functions)에서 발생할 가능성이 높기 때문에, L2CAP 상태를 많이 커버할수록 취약점 탐지 가능성이 높아질 수 있다. 상태 커버리지는 프로토콜 리버스 엔지니어링 도구(protocol reverse engineering tool)를 통해 측정될 수 있다.

[0162] L2Fuzz, Defensics, BFuzz 및 BSS를 각각 이용하여 퍼징 테스트를 수행한 실험 결과는 도 4를 참조하여 후술한다.

[0164] 도 4는 본 개시의 몇몇 실시예에 따른 퍼징을 수행하기 위한 컴퓨팅 장치의 프로그램과 종래의 기술들의 상태 커버리지(state coverage)를 나타낸 도면이다.

[0165] 도 4의 실험 결과에 따른 상태 커버리지를 참조하면, L2Fuzz는 13개의 상태를 커버할 수 있다. 종래 기술인 Defensics은 7개의 상태를 커버할 수 있다. 종래 기술인 BFuzz는 6개의 상태를 커버할 수 있다. 종래의 기술인 BSS는 3개의 상태를 커버할 수 있다. 따라서, 본 개시의 몇몇 실시예에 따른 퍼징을 수행하기 위한 컴퓨팅 장치(100)에 의해 생성 및 저장된 프로그램인 L2Fuzz는 종래의 기술들에 비해서 더 많은 상태를 커버함으로써, 취약점 탐지 가능성이 종래의 기술들에 비해서 더 높다는 것을 알 수 있다.

[0166] 실험 결과에 따른 변형 효율성은 아래의 표 4와 같을 수 있다.

표 4

[0167]

Fuzzer	MP Ratio	PR Ratio	Mutation efficiency
L2Fuzz	69.96%	32.49%	47.22%
Defensics	2.38%	1.73%	2.33%
BFuzz	1.50%	91.60%	0.12%
BSS	0%	0%	0%

[0168] 표 4를 참조하면, L2Fuzz의 변형 효율성은 47.22%일 수 있다. 종래 기술인 Defensics의 변형 효율성은 2.33%일 수 있다. 종래 기술인 BFuzz의 변형 효율성은 0.12%일 수 있다. 종래의 기술인 BSS의 변형 효율성은 0%일 수 있다. 따라서, 본 개시의 몇몇 실시예에 따른 퍼징을 수행하기 위한 컴퓨팅 장치(100)에 의해 생성 및 저장된 프로그램인 L2Fuzz는 종래의 기술들에 비해서 더 높은 변형 효율성을 가짐으로써, 공격의 효율성이 종래의 기술들에 비해서 더 높다는 것을 알 수 있다.

[0170] 도 5는 본 개시의 몇몇 실시예에 따라 퍼징을 수행하기 위한 방법을 예시적으로 나타낸 도면이다.

[0171] 프로세서(110)는 디바이스(200)에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 상기 적어도 하나의 포트 중에서 제 1 포트를 선택할 수 있다(S110).

[0172] 프로세서(110)는 적어도 하나의 포트 중에서 페어링을 요구하고, 그리고 페어링의 요구를 무시하고 통신이 가능한 제 1 포트를 선택할 수 있다.

[0173] 프로세서(110)는 적어도 하나의 포트 중에서 페어링을 요구하지 않는 제 1 포트를 선택할 수 있다.

[0174] 프로세서(110)는 제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 제 1 포트에 전송할 수 있다(S120).

[0175] 프로세서(110)는 제 1 상태 전이 패킷을 제 1 포트에 전송(S120)하기 이전에, 작업(job)에 기초하여 디바이스(200)의 통신 프로토콜에 따라 사전 결정된 복수의 상태들을 복수의 그룹들로 분류할 수 있다. 프로세서(110)는 복수의 그룹들 각각에 대응되는 명령어를 매핑시킬 수 있다.

[0176] 프로세서(110)는 제 1 포트로부터 수신한 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 제 1 포트의 현재 상태(state)(예를 들어, 제 1 현재 상태)를 결정할 수 있다(S130).

[0177] 프로세서(110)는 제 1 포트의 현재 상태(예를 들어, 제 1 현재 상태)가 사전 결정된 제 1 상태에 대응되는

경우, 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성할 수 있다(S140).

[0178] 명령어에 포함된 적어도 하나의 필드는 고정되어 변경 불가능한 데이터를 포함하는 고정 필드(fixed field), 다른 필드에 종속된 데이터를 포함하는 종속 필드(dependent field) 및 변형 가능한 데이터를 포함하는 변형 필드(mutable field)를 포함할 수 있다.

[0179] 변형 필드는 디바이스(200)의 통신 채널 또는 포트 중 적어도 하나와 관련된 데이터를 포함하는 핵심 필드(core field) 및 응용 소프트웨어와 관련된 데이터를 포함하는 응용 필드(application field)를 포함할 수 있다.

[0180] 프로세서(110)는 제 1 포트의 제 1 현재 상태가 사전 결정된 제 1 상태에 대응되는 경우, 명령어에 포함된 핵심 필드를 변형시킨 제 1 테스트 패킷을 생성할 수 있다.

[0181] 프로세서(110)는 제 1 테스트 패킷을 이용하여 제 1 포트의 퍼징 테스트를 수행할 수 있다(S150).

[0182] 프로세서(110)는 제 1 테스트 패킷을 제 1 포트에 전송할 수 있다. 프로세서(110)는 제 1 포트로부터 수신한 제 1 테스트 패킷에 대한 제 2 응답 패킷에 기초하여 제 1 포트의 취약점 발현 여부를 결정할 수 있다.

[0183] 프로세서(110)는 제 2 응답 패킷과 사전 저장된 정상 응답 패킷이 서로 상이한 경우, 제 1 포트에 제 1 테스트 패킷에 의한 취약점이 발현된 것으로 결정할 수 있다. 프로세서(110)는 발현된 취약점에 관한 정보를 생성할 수 있다.

[0184] 프로세서(110)는 제 1 포트를 제 1 상태와 상이한 사전 결정된 제 2 상태(state)로 진입시키기 위해, 사전 결정된 제 2 상태에 대응되는 명령어가 매핑(mapping)된 제 2 상태 전이 패킷을 제 1 포트에 전송할 수 있다.

[0185] 프로세서(110)는 제 1 포트로부터 수신한 제 2 상태 전이 패킷에 대한 제 3 응답 패킷에 기초하여 제 1 포트의 제 2 현재 상태(state)를 결정할 수 있다.

[0186] 프로세서(110)는 제 1 포트의 제 2 현재 상태가 사전 결정된 제 2 상태에 대응되는 경우, 사전 결정된 제 2 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 2 테스트 패킷을 생성할 수 있다.

[0187] 프로세서(110)는 제 2 테스트 패킷을 이용하여 제 1 포트의 퍼징 테스트를 수행할 수 있다.

[0188] 도 5에 도시되는 단계들은 예시적인 단계들이다. 따라서, 본 개시내용의 사상의 범위를 벗어나지 않는 한도에서도 5의 단계들 중 일부가 생략되거나 추가적인 단계들이 존재할 수 있다는 점 또한 당업자에게 명백할 것이다. 또한, 도 5에 기재된 구성들(예를 들어, 컴퓨팅 장치(100), 디바이스(200)등)에 관한 구체적인 내용은 앞서 도 1 내지 도 4를 통해 설명한 내용으로 대체될 수 있다.

[0189] 도 1 내지 도 5을 참조하여 상술한 바와 같이, 본 개시에 따른 퍼징을 수행하기 위한 컴퓨팅 장치(100)는 테스트의 타겟이 되는 포트와 페어링 없이 테스트를 진행함으로써, 페어링 없이 공격이 가능한 패킷을 생성 및 탐지할 수 있다.

[0190] 또한, 본 개시에 따른 퍼징을 수행하기 위한 컴퓨팅 장치(100)는 포트의 상태를 변경시키고, 포트의 상태에 대응되는 명령어에서 핵심 필드를 중심으로 변형시킨 테스트 패킷을 생성함으로써, 포트의 취약점 발현 가능성을 높일 수 있다.

[0191] 또한, 본 개시에 따른 퍼징을 수행하기 위한 컴퓨팅 장치(100)는 테스트 패킷에 의해 발현된 취약점에 관한 정보를 생성하여 저장 또는 취약점이 발현된 포트의 디바이스(200)에 전송함으로써, 컴퓨팅 장치(100) 또는 디바이스(200)의 사용자가 취약점에 관한 정보를 확인하고 대처 방안을 강구할 수 있다.

[0193] 도 6은 본 개시의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도이다.

[0194] 본 개시가 일반적으로 컴퓨팅 장치에 의해 구현될 수 있는 것으로 전술되었지만, 당업자라면 본 개시가 하나 이상의 컴퓨터 상에서 실행될 수 있는 컴퓨터 실행가능 명령어 및/또는 기타 프로그램 모듈들과 결합되어 및/또는 하드웨어와 소프트웨어의 조합으로써 구현될 수 있다는 것을 잘 알 것이다.

[0195] 일반적으로, 프로그램 모듈은 특정의 태스크를 수행하거나 특정의 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조, 기타 등등을 포함한다. 또한, 당업자라면 본 개시의 방법이 단일-프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터는 물론 퍼스널 컴퓨터, 핸드헬드(handheld) 컴퓨팅 장치, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 기타 등등(이들 각각은 하나 이상의 연관된

장치와 연결되어 동작할 수 있음)을 비롯한 다른 컴퓨터 시스템 구성으로 실시될 수 있다는 것을 잘 알 것이다.

[0196] 본 개시의 설명된 실시예들은 또한 어떤 태스크들이 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘 다에 위치할 수 있다.

[0197] 컴퓨터는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 일시적(transitory) 및 비일시적(non-transitory) 매체, 이동식 및 비-이동식 매체를 포함한다. 제한이 아닌 예로서, 컴퓨터 판독가능 매체는 컴퓨터 판독가능 저장 매체 및 컴퓨터 판독가능 전송 매체를 포함할 수 있다. 컴퓨터 판독가능 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성 매체, 일시적 및 비-일시적 매체, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital video disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함하지만, 이에 한정되지 않는다.

[0198] 컴퓨터 판독가능 전송 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터 등을 구현하고 모든 정보 전달 매체를 포함한다. 피변조 데이터 신호라는 용어는 신호 내에 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 제한이 아닌 예로서, 컴퓨터 판독가능 전송 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들 중 임의의 것의 조합도 역시 컴퓨터 판독가능 전송 매체의 범위 안에 포함되는 것으로 한다.

[0199] 컴퓨터(1102)를 포함하는 본 개시의 여러가지 측면들을 구현하는 예시적인 환경(1100)이 나타내어져 있으며, 컴퓨터(1102)는 처리 장치(1104), 시스템 메모리(1106) 및 시스템 버스(1108)를 포함한다. 시스템 버스(1108)는 시스템 메모리(1106)(이에 한정되지 않음)를 비롯한 시스템 컴포넌트들을 처리 장치(1104)에 연결시킨다. 처리 장치(1104)는 다양한 상용 프로세서들 중 임의의 프로세서일 수 있다. 듀얼 프로세서 및 기타 멀티프로세서 아키텍처도 역시 처리 장치(1104)로서 이용될 수 있다.

[0200] 시스템 버스(1108)는 메모리 버스, 주변장치 버스, 및 다양한 상용 버스 아키텍처 중 임의의 것을 사용하는 로컬 버스에 추가적으로 상호 연결될 수 있는 몇 가지 유형의 버스 구조 중 임의의 것일 수 있다. 시스템 메모리(1106)는 판독 전용 메모리(ROM)(1110) 및 랜덤 액세스 메모리(RAM)(1112)를 포함한다. 기본 입/출력 시스템(BIOS)은 ROM, EPROM, EEPROM 등의 비휘발성 메모리(1110)에 저장되며, 이 BIOS는 시동 중과 같은 때에 컴퓨터(1102) 내의 구성요소들 간에 정보를 전송하는 일을 돕는 기본적인 루틴을 포함한다. RAM(1112)은 또한 데이터를 캐싱하기 위한 정적 RAM 등의 고속 RAM을 포함할 수 있다.

[0201] 컴퓨터(1102)는 또한 내장형 하드 디스크 드라이브(HDD)(1114)(예를 들어, EIDE, SATA)-이 내장형 하드 디스크 드라이브(1114)는 또한 적당한 섀시(도시 생략) 내에서 외장형 용도로 구성될 수 있음-, 자기 플로피 디스크 드라이브(FDD)(1116)(예를 들어, 이동식 디스켓(1118)으로부터 판독을 하거나 그에 기록을 하기 위한 것임), 및 광 디스크 드라이브(1120)(예를 들어, CD-ROM 디스크(1122)를 판독하거나 DVD 등의 기타 고용량 광 매체로부터 판독을 하거나 그에 기록을 하기 위한 것임)를 포함한다. 하드 디스크 드라이브(1114), 자기 디스크 드라이브(1116) 및 광 디스크 드라이브(1120)는 각각 하드 디스크 드라이브 인터페이스(1124), 자기 디스크 드라이브 인터페이스(1126) 및 광 드라이브 인터페이스(1128)에 의해 시스템 버스(1108)에 연결될 수 있다. 외장형 드라이브 구현을 위한 인터페이스(1124)는 USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술 중 적어도 하나 또는 그 둘 다를 포함한다.

[0202] 이들 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어, 기타 등등의 비휘발성 저장을 제공한다. 컴퓨터(1102)의 경우, 드라이브 및 매체는 임의의 데이터를 적당한 디지털 형식으로 저장하는 것에 대응한다. 상기에서의 컴퓨터 판독가능 매체에 대한 설명이 HDD, 이동식 자기 디스크, 및 CD 또는 DVD 등의 이동식 광 매체를 언급하고 있지만, 당업자라면 집 드라이브(zip drive), 자기 카세트, 플래쉬 메모리 카드, 카트리지, 기타 등등의 컴퓨터에 의해 판독가능한 다른 유형의 매체도 역시 예시적인 운영 환경에서 사용될 수 있으며 또 임의의 이러한 매체가 본 개시의 방법들을 수행하기 위한 컴퓨터 실행가능 명령어를 포함할 수 있다는 것을 잘 알 것이다.

[0203] 운영 체제(1130), 하나 이상의 애플리케이션 프로그램(1132), 기타 프로그램 모듈(1134) 및 프로그램 데이터(1136)를 비롯한 다수의 프로그램 모듈이 드라이브 및 RAM(1112)에 저장될 수 있다. 운영 체제, 애플리케이션, 모듈 및/또는 데이터의 전부 또는 그 일부분이 또한 RAM(1112)에 캐싱될 수 있다. 본 개시가 여러가지 상업적으로 이용가능한 운영 체제 또는 운영 체제들의 조합에서 구현될 수 있다는 것을 잘 알 것이다.

[0204] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들어, 키보드(1138) 및 마우스(1140) 등의 포인팅 장치를 통해 컴퓨터(1102)에 명령 및 정보를 입력할 수 있다. 기타 입력 장치(도시 생략)로는 마이크, IR 리모콘, 조이스틱, 게임 패드, 스타일러스 펜, 터치 스크린, 기타 등등이 있을 수 있다. 이들 및 기타 입력 장치가 종종 시스템 버스(1108)에 연결되어 있는 입력 장치 인터페이스(1142)를 통해 처리 장치(1104)에 연결되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스, 기타 등등의 기타 인터페이스에 의해 연결될 수 있다.

[0205] 모니터(1144) 또는 다른 유형의 디스플레이 장치도 역시 비디오 어댑터(1146) 등의 인터페이스를 통해 시스템 버스(1108)에 연결된다. 모니터(1144)에 부가하여, 컴퓨터는 일반적으로 스피커, 프린터, 기타 등등의 기타 주변 출력 장치(도시 생략)를 포함한다.

[0206] 컴퓨터(1102)는 유선 및/또는 무선 통신을 통한 원격 컴퓨터(들)(1148) 등의 하나 이상의 원격 컴퓨터로의 논리적 연결을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(1148)는 워크스테이션, 컴퓨팅 디바이스 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서-기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있으며, 일반적으로 컴퓨터(1102)에 대해 기술된 구성요소들 중 다수 또는 그 전부를 포함하지만, 간략함을 위해, 메모리 저장 장치(1150)만이 도시되어 있다. 도시되어 있는 논리적 연결은 근거리 통신망(LAN)(1152) 및/또는 더 큰 네트워크, 예를 들어, 원거리 통신망(WAN)(1154)에의 유선/무선 연결을 포함한다. 이러한 LAN 및 WAN 네트워킹 환경은 사무실 및 회사에서 일반적인 것이며, 인트라넷 등의 전사적 컴퓨터 네트워크(enterprise-wide computer network)를 용이하게 해주며, 이들 모두는 전세계 컴퓨터 네트워크, 예를 들어, 인터넷에 연결될 수 있다.

[0207] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(1156)를 통해 로컬 네트워크(1152)에 연결된다. 어댑터(1156)는 LAN(1152)에의 유선 또는 무선 통신을 용이하게 해줄 수 있으며, 이 LAN(1152)은 또한 무선 어댑터(1156)와 통신하기 위해 그에 설치되어 있는 무선 액세스 포인트를 포함하고 있다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 모뎀(1158)을 포함할 수 있거나, WAN(1154) 상의 통신 컴퓨팅 디바이스에 연결되거나, 또는 인터넷을 통하는 등, WAN(1154)을 통해 통신을 설정하는 기타 수단을 갖는다. 내장형 또는 외장형 및 유선 또는 무선 장치일 수 있는 모뎀(1158)은 직렬 포트 인터페이스(1142)를 통해 시스템 버스(1108)에 연결된다. 네트워크화된 환경에서, 컴퓨터(1102)에 대해 설명된 프로그램 모듈들 또는 그의 일부분이 원격 메모리/저장 장치(1150)에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이며 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 잘 알 것이다.

[0208] 컴퓨터(1102)는 무선 통신으로 배치되어 동작하는 임의의 무선 장치 또는 개체, 예를 들어, 프린터, 스캐너, 데스크톱 및/또는 휴대용 컴퓨터, PDA(portable data assistant), 통신 위성, 무선 검출가능 태그와 연관된 임의의 장비 또는 장소, 및 전화와 통신을 하는 동작을 한다. 이것은 적어도 Wi-Fi 및 블루투스 무선 기술을 포함한다. 따라서, 통신은 종래의 네트워크에서와 같이 미리 정의된 구조이거나 단순하게 적어도 2개의 장치 사이의 애드혹 통신(ad hoc communication)일 수 있다.

[0209] Wi-Fi(Wireless Fidelity)는 유선 없이도 인터넷 등으로의 연결을 가능하게 해준다. Wi-Fi는 이러한 장치, 예를 들어, 컴퓨터가 실내에서 및 실외에서, 즉 기지국의 통화권 내의 아무 곳에서나 데이터를 전송 및 수신할 수 있게 해주는 셀 전화와 같은 무선 기술이다. Wi-Fi 네트워크는 안전하고 신뢰성 있으며 고속인 무선 연결을 제공하기 위해 IEEE 802.11(a, b, g, 기타)이라고 하는 무선 기술을 사용한다. 컴퓨터를 서로에, 인터넷에 및 유선 네트워크(IEEE 802.3 또는 이더넷을 사용함)에 연결시키기 위해 Wi-Fi가 사용될 수 있다. Wi-Fi 네트워크는 비인가 2.4 및 5GHz 무선 대역에서, 예를 들어, 11Mbps(802.11a) 또는 54 Mbps(802.11b) 데이터 레이트로 동작하거나, 양 대역(듀얼 대역)을 포함하는 제품에서 동작할 수 있다.

[0210] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 정보 및 신호들이 임의의 다양한 상이한 기술들 및 기법들을 이용하여 표현될 수 있다는 것을 이해할 것이다. 예를 들어, 위의 설명에서 참조될 수 있는 데이터, 지시들, 명령들, 정보, 신호들, 값들, 심볼들 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 입자들, 광학장들 또는 입자들, 또는 이들의 임의의 결합에 의해 표현될 수 있다.

[0211] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 프로세서들, 수단들, 회로들 및 알고리즘 단계들이 전자 하드웨어, (편의를 위해, 여기에서 소프트웨어로 지칭되는) 다양한 형태들의 프로그램 또는 설계 코드 또는 이들 모두의 결합에 의해 구현될 수 있다는 것을 이해할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 호환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 이들의 기능과 관련하여 위에서 일반적으로 설명되었다. 이러한 기능이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 애플리케이션 및 전체 시스템에 대하여 부과되는 설계 제약들에 따라 좌우된다. 본 개시의 기술 분야에서 통상의 지식을 가진 자는 각각의 특정한 애플리케이션에 대하여 다양한 방식들로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 개시의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.

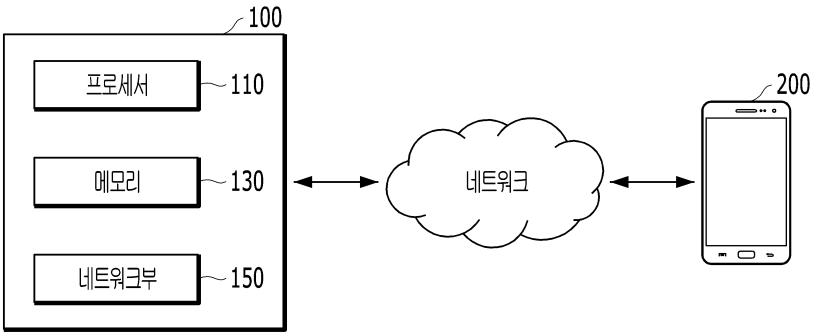
[0212] 여기서 제시된 다양한 실시예들은 방법, 장치, 또는 표준 프로그래밍 및/또는 엔지니어링 기술을 사용한 제조 물품(article)으로 구현될 수 있다. 용어 제조 물품은 임의의 컴퓨터-판독가능 저장장치로부터 액세스 가능한 컴퓨터 프로그램, 캐리어, 또는 매체(media)를 포함한다. 예를 들어, 컴퓨터-판독가능 저장매체는 자기 저장 장치(예를 들면, 하드 디스크, 플로피 디스크, 자기 스트립, 등), 광학 디스크(예를 들면, CD, DVD, 등), 스마트 카드, 및 플래쉬 메모리 장치(예를 들면, EEPROM, 카드, 스틱, 키 드라이브, 등)를 포함하지만, 이들로 제한되는 것은 아니다. 또한, 여기서 제시되는 다양한 저장 매체는 정보를 저장하기 위한 하나 이상의 장치 및/또는 다른 기계-판독가능한 매체를 포함한다.

[0213] 제시된 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조는 예시적인 접근들의 일례임을 이해하도록 한다. 설계 우선순위들에 기반하여, 본 개시의 범위 내에서 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조가 재배열될 수 있다는 것을 이해하도록 한다. 첨부된 방법 청구항들은 샘플 순서로 다양한 단계들의 엘리먼트들을 제공하지만 제시된 특정한 순서 또는 계층 구조에 한정되는 것을 의미하지는 않는다.

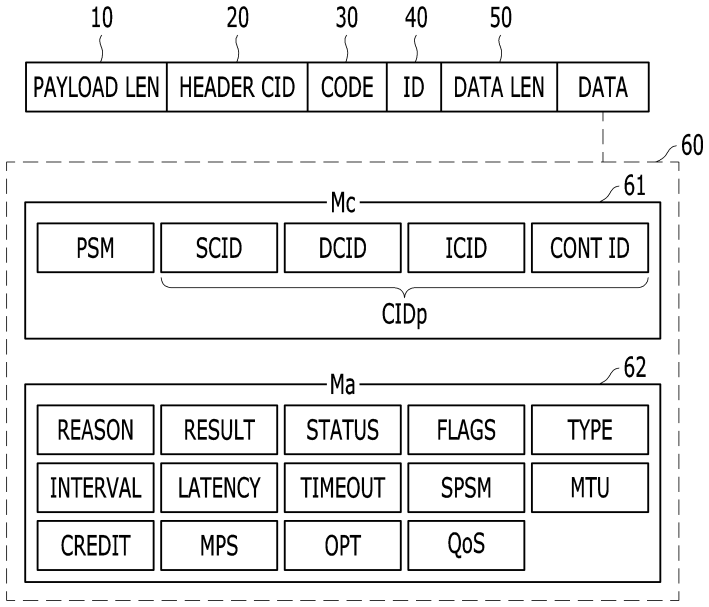
[0214] 제시된 실시예들에 대한 설명은 임의의 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 개시를 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

도면

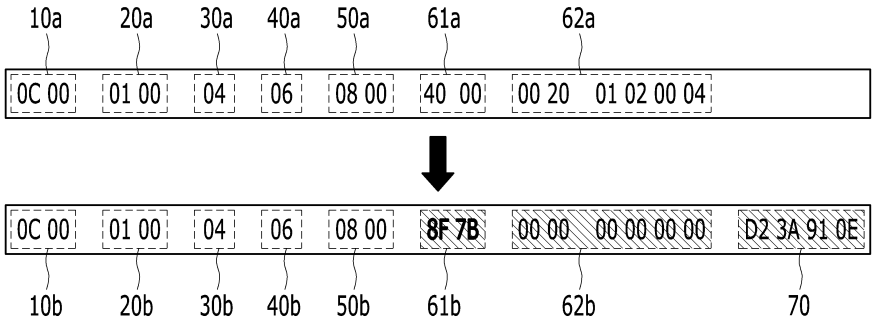
도면1



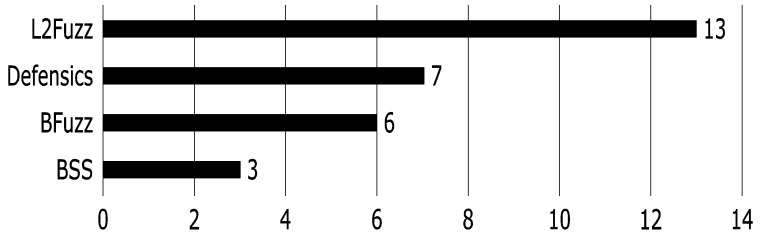
도면2


10
20
30
40
50
PAYLOAD LEN
HEADER CID
CODE
ID
DATA LEN
DATA
60
61
Mc
PSM
SCID
DCID
ICID
CONT ID
CIDp
62
Ma
REASON
RESULT
STATUS
FLAGS
TYPE
INTERVAL
LATENCY
TIMEOUT
SPSM
MTU
CREDIT
MPS
OPT
QoS

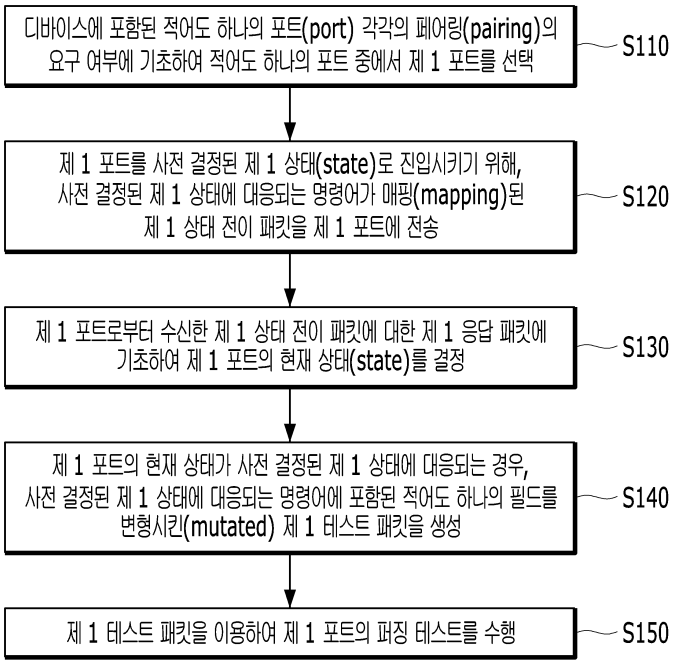
도면3


10a
20a
30a
40a
50a
61a
62a
0C 00
01 00
04
06
08 00
40 00
00 20
01 02 00 04
0C 00
01 00
04
06
08 00
8F 7B
00 00
00 00 00 00
D2 3A 91 0E
10b
20b
30b
40b
50b
61b
62b
70

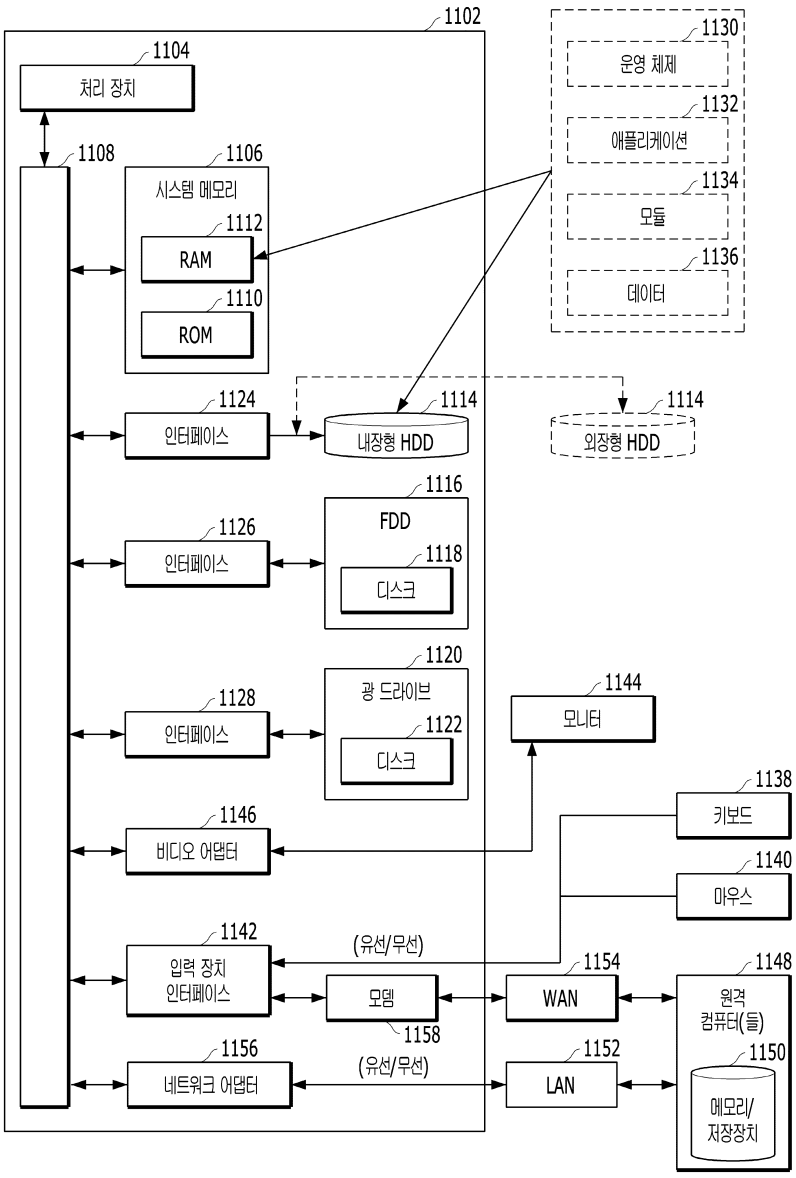
도면4


L2Fuzz
13
Defensics
7
BFuzz
6
BSS
3
0
2
4
6
8
10
12
14

도면5


디바이스에 포함된 적어도 하나의 포트(port) 각각의 페어링(pairing)의 요구 여부에 기초하여 적어도 하나의 포트 중에서 제 1 포트를 선택
S110
제 1 포트를 사전 결정된 제 1 상태(state)로 진입시키기 위해, 사전 결정된 제 1 상태에 대응되는 명령어가 매핑(mapping)된 제 1 상태 전이 패킷을 제 1 포트에 전송
S120
제 1 포트로부터 수신한 제 1 상태 전이 패킷에 대한 제 1 응답 패킷에 기초하여 제 1 포트의 현재 상태(state)를 결정
S130
제 1 포트의 현재 상태가 사전 결정된 제 1 상태에 대응되는 경우, 사전 결정된 제 1 상태에 대응되는 명령어에 포함된 적어도 하나의 필드를 변형시킨(mutated) 제 1 테스트 패킷을 생성
S140
제 1 테스트 패킷을 이용하여 제 1 포트의 퍼징 테스트를 수행
S150

도면6


1102
1104
처리 장치
1108
1106
시스템 메모리
1112
RAM
1110
ROM
1130
운영 체제
1132
애플리케이션
1134
모듈
1136
데이터
1124
인터페이스
1114
내장형 HDD
1114
외장형 HDD
1116
FDD
1126
인터페이스
1118
디스크
1120
광 드라이브
1128
인터페이스
1122
디스크
1144
모니터
1138
키보드
1146
비디오 어댑터
1140
마우스
1142
입력 장치 인터페이스
(유선/무선)
1154
WAN
1148
원격 컴퓨터(들)
모뎀
1158
1156
네트워크 어댑터
(유선/무선)
1152
LAN
1150
메모리/ 저장장치