(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년05월29일
(11) 등록번호 10-2104610
(24) 등록일자 2020년04월20일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01)
(52) CPC특허분류
H04L 63/1433 (2013.01)
(21) 출원번호 10-2018-0028553
(22) 출원일자 2018년03월12일
심사청구일자 2018년03월12일
(65) 공개번호 10-2019-0107373
(43) 공개일자 2019년09월20일
(56) 선행기술조사문헌
KR1020100072707 A*
US20130340083 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
주식회사 아이오티큐브

(72) 발명자
이희조

김동혁

(뒷면에 계속)

(74) 대리인
특허법인엠에이피에스

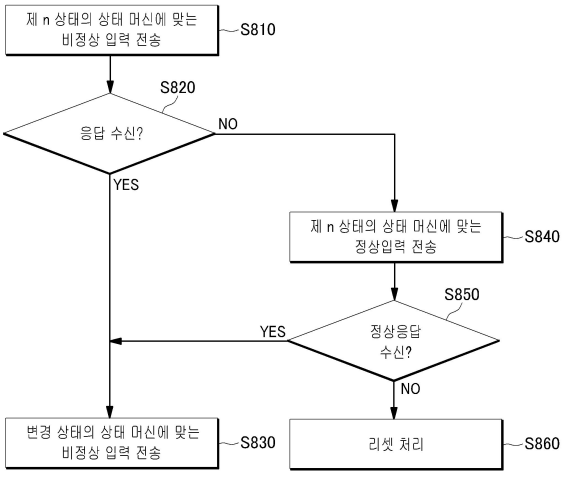
전체 청구항 수 : 총 4 항 심사관 : 문형섭

(54) 발명의 명칭 **네트워크 프로토콜의 취약점을 탐지하는 퍼징 방법 및 장치**

(57) 요 약

본 발명의 네트워크 프로토콜의 취약점을 탐지하는 퍼징 방법은 (a) 취약점 확인 대상 장치의 유한 상태 머신에 대하여, 제 n 상태(n은 자연수)의 유한 상태 머신에 대하여 해당 상태에 맞게 설정된 비정상 입력을 전송하는 단계; 및 (b) 상기 제 n 상태의 유한 상태 머신으로부터 응답을 수신하는 경우, 해당 응답이 나타내는 상기 유한 상태 머신의 변경 상태에 맞는 비정상 입력을 전송하고, 상기 제 n 상태의 유한 상태 머신으로부터 제한 시간내에 응답을 수신하지 못하는 경우 리셋 처리를 통해 최초 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행하는 단계를 포함한다.

대 표 도 - 도8



(72) 발명자 **한지연**

김상우

이 발명을 지원한 국가연구개발사업

과제고유번호	1711055304
부처명	과학기술정보통신부
연구관리전문기관	정보통신기술진흥센터
연구사업명	정보보호핵심원천기술개발(R&D)
연구과제명	(자가방어-2세부) 자기학습형 사이버면역 기술개발
기 여 율	1/1
주관기관	한국인터넷진흥원
연구기간	2017.03.01 ~ 2017.12.31

명 세 서

청구범위

청구항 1

네트워크 프로토콜의 취약점을 탐지하는 퍼징 방법에 있어서,

(a) 취약점 확인 대상 장치의 유한 상태 머신에 대하여, 제 n 상태(n은 자연수)의 유한 상태 머신에 대하여 해당 상태에 맞게 설정된 비정상 입력을 전송하는 단계; 및

(b) 상기 비정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 정상 응답을 수신하는 경우, 상기 유한 상태 머신이 상기 정상 응답이 나타내는 변경 상태로 전이한 것으로 간주하여 변경 상태에 맞는 비정상 입력을 전송하는 단계;

(c) 상기 제 n 상태의 유한 상태 머신으로부터 상기 비정상 입력을 거부하는 실패 응답을 수신하는 경우, 상기 유한 상태 머신이 최초 상태로 상태 전이가 이루어진 것으로 보고, 해당 상태에 맞는 비정상 입력을 전송하는 단계

(d) 상기 비정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 응답을 제한 시간 내에 수신하지 못한 경우, 상기 제 n 상태의 유한 상태 머신에 대하여 상기 제 n 상태에 맞는 정상 입력을 전송하는 단계;

(e) 상기 정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 정상 응답을 수신한 경우, 상기 유한 상태 머신이 상기 정상 응답이 나타내는 변경 상태로 전이한 것으로 간주하여 변경 상태에 맞는 비정상 입력을 전송하는 단계; 및

(f) 상기 정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 정상 응답을 수신하지 못한 경우, 리셋 처리를 통해 초기 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행하는 단계를 포함하되,

상기 제한 시간은 상기 유한 상태 머신의 각 상태별로 상태 전이에 소요되는 시간에 기초하여 설정된 가중치에 기초하여 설정되는 것인는 퍼징 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

제 1 항에 있어서,

상기 (b) 단계 또는 (c) 단계는 상기 확인 대상 장치가 상기 비정상 입력에 대하여 취약하지 않음을 기록하는 것인 퍼징 방법.

청구항 5

삭제

청구항 6

제 1 항에 있어서,

상기 취약점 확인 대상 장치의 유한 상태 머신에 대하여 각 상태별로 상태 전이를 위해 입력을 전송할 때 응답이 수신되는데 소요되는 평균 시간을 측정하고, 상기 평균 시간에 기초하여 상태 전이 별로 상기 가중치를 설정하는 단계를 더 포함하는 퍼징 방법.

청구항 7

네트워크 프로토콜의 취약점을 탐지하는 퍼징 장치에 있어서,

네트워크 프로토콜의 취약점을 탐지하는 프로그램이 저장된 메모리,

데이터 입출력을 수행하는 데이터 입출력 모듈 및

프로세서를 포함하되,

상기 프로그램은 상기 프로세서에 의하여 구동되어, 취약점 확인 대상 장치의 제 n 상태(n은 자연수)의 유한 상태 머신에 대하여 해당 상태에 맞게 설정된 비정상 입력을 전송하고, 상기 비정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 정상 응답을 수신하는 경우, 상기 유한 상태 머신이 상기 정상 응답이 나타내는 변경 상태로 전이한 것으로 간주하여 변경 상태에 맞는 비정상 입력을 전송하고, 상기 제 n 상태의 유한 상태 머신으로부터 상기 비정상 입력을 거부하는 실패 응답을 수신하는 경우, 상기 유한 상태 머신이 최초 상태로 상태 전이가 이루어진 것으로 보고, 해당 상태에 맞는 비정상 입력을 전송하는, 상기 비정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 응답을 제한 시간 내에 수신하지 못한 경우, 상기 제 n 상태의 유한 상태 머신에 대하여 상기 제 n 상태에 맞는 정상 입력을 전송하고, 상기 정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 정상 응답을 수신한 경우, 상기 유한 상태 머신이 상기 정상 응답이 나타내는 변경 상태로 전이한 것으로 간주하여 변경 상태에 맞는 비정상 입력을 전송하고, 상기 정상 입력에 대하여 상기 제 n 상태의 유한 상태 머신으로부터 정상 응답을 수신하지 못한 경우, 리셋 처리를 통해 초기 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행하되,

상기 제한 시간은 상기 유한 상태 머신의 각 상태별로 상태 전이에 소요되는 시간에 기초하여 설정된 가중치에 기초하여 설정되는 것인 퍼징 장치.

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 네트워크 프로토콜의 취약점을 탐지하는 퍼징 방법 및 장치에 대한 것으로, 보다 상세하게는 프로토콜 퍼징 중 상태 전이 과정을 좀 더 효율적으로 진행해 취약점을 빠르게 점검하는 방법에 관한 것이다.

배 경 기 술

[0002] 최근 들어 스마트 디바이스와 IoT기기 등 인터넷과 연결되는 기기들의 확산으로 인해 소프트웨어의 공격 요소들이 증가하고 있으며, 이에 따라 기기 및 소프트웨어의 취약점을 찾는 것은 갈수록 중요해지고 있다. 현재 취약점을 찾는 여러 가지 방법 중 많이 사용하는 방법으로 퍼징(Fuzzing)이라는 방법이 알려져 있으며, 이는 블랙박스 기반 취약점 탐색 기술로 알려져 있다. 이 기술은 컴퓨터가 랜덤을 포함한 여러 방법으로 만들어낸 예측하기 힘든 비정상 입력(Input) 값을 기기에 주고, 기기가 해당 값을 취약점 없이 잘 처리하는지 확인하는 방식으로 이루어진다. 해당 기법은 다른 기법들에 비해 구현하기가 상대적으로 쉽고, 완전 자동화가 가능하므로 충분한 컴퓨팅 자원을 투입하면 보다 쉽게 취약점을 찾을 수 있다.

[0003] 또한 퍼징은 프로토콜을 대상으로도 적용할 수 있으며, 이를 프로토콜 퍼징이라고 한다. 프로토콜 퍼징의 경우 일반 퍼징과는 다르게 입력을 주고받는 과정을 고려하여 취약점을 탐지한다. 프로토콜을 구현한 프로그램은 입출력에 따라 현재 상태를 전이(State transition)하며 동작한다. 이는 이전 입력에 따라 현재의 상태가 정해진다는 뜻이다. 또한 각각의 상태는 다른 취약점을 가질 가능성이 있으므로 퍼저(Fuzzer)는 대상 기기의 상태를 전이시키며 취약점을 확인해야 한다.

[0004] 하지만 해당 방법은 근본적인 한계점을 갖고 있다. 초기의 퍼징 방법은 완전한 랜덤 입력 값을 입력하여 취약점을 찾는 방식이었다. 하지만 입력 값 경우의 수가 길이에 따라 지수함수로 증가하기 때문에 모든 것을 테스트하는 것은 불가능하다. 따라서 현존 기술들은 매우 많은 입력 경우의 수에서 취약점을 발견할 확률이 높은 입력을 선별하는 기술에 초점을 맞추고 있으며, 이로 인해 취약점을 발견하기 위해 상당한 리소스 낭비가 필요하다는 단점이 있다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 대한민국등록특허 제 10-1689795호(통신 프로토콜 소프트웨어의 취약성 검출 방법 및 시스템)

발명의 내용

해결하려는 과제

[0006] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, 본 발명은 네트워크 프로토콜의 취약점을 탐지하기 위한 퍼징 장치 및 방법을 제공하는 것을 그 목적으로 한다.

[0007] 다만, 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0008] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 일측면에 따른 네트워크 프로토콜의 취약점을 탐지하는 퍼징 방법은 (a) 취약점 확인 대상 장치의 유한 상태 머신에 대하여, 제 n 상태(n은 자연수)의 유한 상태 머신에 대하여 해당 상태에 맞게 설정된 비정상 입력을 전송하는 단계; 및 (b) 상기 제 n 상태의 유한 상태 머신으로부터 응답을 수신하는 경우, 해당 응답이 나타내는 상기 유한 상태 머신의 변경 상태에 맞는 비정상 입력을 전송하고, 상기 제 n 상태의 유한 상태 머신으로부터 제한 시간내에 응답을 수신하지 못하는 경우 리셋 처리를 통해 최초 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행하는 단계를 포함한다.

[0009] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 일측면에 따른 네트워크 프로토콜의 취약점을 탐지하는 퍼징 장치는 네트워크 프로토콜의 취약점을 탐지하는 프로그램이 저장된 메모리, 데이터 입출력을 수행하는 데이터 입출력 모듈 및 프로세서를 포함하되, 상기 프로그램은 상기 프로세서에 의하여 구동되어, 취약점 확인 대상 장치의 제 n 상태(n은 자연수)의 유한 상태 머신에 대하여 해당 상태에 맞게 설정된 비정상 입력을 전송하고, 상기 제 n 상태의 유한 상태 머신으로부터 응답을 수신하는 경우, 해당 응답이 나타내는 상기 유한 상태 머신의 변경 상태에 맞는 비정상 입력을 전송하고, 상기 제 n 상태의 유한 상태 머신으로부터 제한 시간내에 응답을 수신하지 못하는 경우 리셋 처리를 통해 최초 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행한다.

발명의 효과

[0010] 전술한 본 발명의 과제 해결 수단에 의하면, 이러한 문제를 해결하고자, 프로토콜을 사용하는 기기에 대하여 상태를 전이하며 취약점을 찾아내는 알고리즘을 제안하였으며, 이를 통해 종래 알려진 방법에 비하여 보다 빠르게 취약점 탐색이 가능하다.

도면의 간단한 설명

[0011] 도 1은 본 발명의 일 실시예에 따른 퍼징 장치를 도시한 블록도이다.

도 2는 본 발명의 일 실시예에 따른 퍼징 과정에서 유한 상태 머신의 상태 전이 과정을 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 퍼징 과정에서 유한 상태 머신의 가중치를 산출하는 과정을 설명하기 위한 도면이다.

도 4는 본 발명의 일 실시예에 따른 퍼징 과정에서 유한 상태 머신의 전이 과정을 설명하기 위한 도면이다.

도 5는 종래의 퍼징 방법을 나타내는 유사 알고리즘을 도시한 도면이다.

도 6은 본 발명의 일 실시예에 따른 퍼징 방법을 나타내는 유사 알고리즘을 도시한 도면이다.

도 7은 본 발명의 일 실시예에 따른 퍼징 방법에서 비정상 입력을 전송하는 순서를 정하는 방법을 나타내는 유사 알고리즘을 도시한 도면이다

도 8은 본 발명의 일 실시예에 따른 퍼징 방법을 도시한 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0013] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0014] 도 1은 본 발명의 일 실시예에 따른 퍼징 장치를 도시한 블록도이다.

[0015] 도시된 바와 같이, 퍼징 장치(100)는 데이터 입출력모듈(110), 메모리(120), 프로세서(130)를 포함한다.

[0016] 데이터 입출력모듈(110)은 통신 모듈을 통해 데이터를 수신 또는 송신하는 기능을 수행할 수 있다. 데이터 입출력 모듈(110)은 다른 네트워크 장치와 유무선 연결을 통해 제어 신호 또는 데이터 신호와 같은 신호를 송수신하기 위해 필요한 하드웨어 및 소프트웨어를 포함하는 장치일 수 있다.

[0017] 메모리(120)에는 네트워크 프로토콜의 취약점을 탐지하는 프로그램이 저장된다. 해당 프로그램은 프로세서(130)에 의하여 구동되어, 취약점 확인 대상 장치의 유한 상태 머신에 대하여, 제 n 상태(n은 자연수)의 유한 상태 머신에 대하여 해당 상태에 맞게 설정된 비정상 입력을 전송하고, 제 n 상태의 유한 상태 머신으로부터 응답을 수신하는 경우 , 유한 상태 머신에 상태 전이가 발생한 것으로 보고, 해당 응답이 나타내는 유한 상태 머신의 변경상태에 맞는 비정상 입력을 전송한다. 응답으로는 정상 응답 또는 실패 응답이 수신될 수 있다. 이때, 정상 응답은 비정상 입력에도 불구하고 이를 정상 입력과 마찬가지로 처리하는 응답 처리를 나타내고, 실패 응답은 비정상 입력을 거부하는 취지로 출력되는 것을 의미한다.

[0018] 한편, 변경 상태는 프로토콜의 유한 상태 머신에 따라 달라질 수 있으며, 제 n 상태에서 전이된 제 n-1 상태, 제 n+1 상태 또는 제 n+2 상태 등을 의미한다. 만약, 제 n 상태의 유한 상태 머신으로부터 제한 시간내에 응답을 수신하지 못하는 경우 리셋 처리를 통해 최초 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행하는 동작을 수행한다.

[0019] 이러한 메모리(120)는 전원이 공급되지 않아도 저장된 정보를 계속 유지하는 비휘발성 저장장치 또는 저장된 정보를 유지하기 위하여 전력이 필요한 휘발성 저장장치를 통칭하는 것이다.

[0020] 프로세서(130)는 메모리(120)에 저장된 네트워크 프로토콜의 취약점을 탐지하는 프로그램을 수행한다.

[0021] 도 2는 본 발명의 일 실시예에 따른 퍼징 과정에서 유한 상태 머신의 상태 전이 과정을 설명하기 위한 도면이다.

[0022] 취약점 탐지 대상이 되는 프로토콜의 상세한 스펙(specification)에 대해서는 미리 알려져 있는 것으로 가정한다. 그리고, 이러한 스펙은 유한 상태 머신(Fininte state machine)에 의하여 정의된다. 예를 들어, 서로 다른 2 이상의 장치가 특정 프로토콜을 기반으로 통신을 수행하는 경우, 각각의 장치는 해당 프로토콜의 통신을 수행하는 과정에서 신호를 전송하고 이에 대한 응답을 수신하는 각각의 단계 또는 신호를 수신하고 이에 대한 응답을 전송하는 각각의 단계에 따라, 유한 상태 머신의 각 상태가 전이된다.

[0023] 도시된 바와 같은 유한 상태 머신은 서로 다른 4개의 상태(s_0, s_1, s_2, s_3)에서 상태 전이가 이루어 진다. 그리

고, 각각의 상태 머신은 8개의 정보를 가진 튜플($M = (S, s_o, I, O, D, \delta, \gamma, \omega)$)에 의하여 정의될 수 있다.

1) $S = \{s_0, s_1, ..., s_{n-1}\}$: Finite set of all states

2) s_0: An initial state

3) $I = \{i_0, i_1, ..., i_{m-1}\}$: Finite set of inputs including λ

4) $O = \{o_0, o_1, ..., o_{m-1}\}$: Finite set of outputs including λ

5) $D \subseteq S \times I$: Specification domain

6) $\delta : D \rightarrow S$: Transition function

7) $\gamma : D \rightarrow O$: Output function

8) $\omega : D \rightarrow N$: Weight function

[0024]

[0025] 예를 들어, 제 1 상태(s_0)의 유한 상태 머신에 대하여 입력 패킷을 전송하고, 이에 대한 응답이 정상정으로 출력되면, 유한 상태 머신이 제 2 상태(s_1)로 전이된 것으로 본다. 그러나, 이에 대한 응답이 출력되지 않으면, 상태 전이가 이루어지지 않아 제 1 상태(s_0)에 있는 것으로 본다.

[0026] 도 3은 본 발명의 일 실시예에 따른 퍼징 과정에서 유한 상태 머신의 가중치를 산출하는 과정을 설명하기 위한 도면이다.

[0027] 유한 상태 머신에 대하여 각 상태별로 상태 전이에 소요되는 시간에 기초하여 가중치가 설정된다. 프로토콜 퍼징을 통해 취약점을 빠르게 탐지하기 위해서는, 상태 전이 과정에서 소요되는 시간을 고려하여 최단 거리 계산이 필요하다.

[0028] 가중치는 유한 상태 머신이 상태 전이를 일으키는 데 걸리는 시간을 의미하며 이 정보는 어떤 특정 상태로의 가장 빠른 전이 방법을 계산하는 데 필요하다. 가중치는 현재 기기와의 트래픽 상태, 기기의 처리 속도에 따라 결정되며 이는 모든 기기에 대해 일괄적으로 적용할 수 있는 것이 아니다. 따라서 제안한 기법에서는 상태 전이를 위해 입력을 보낼 때 응답이 올 때까지의 시간을 측정하며, 해당 시간의 평균을 상태 전이에 필요한 가중치로 본다.

[0029] 도 4는 본 발명의 일 실시예에 따른 퍼징 과정에서 유한 상태 머신의 전이 과정을 설명하기 위한 도면이다.

[0030] (a)에 도시된 바와 같이, 제 2 상태(s_1)의 상태 머신에 대하여 해당 상태에 맞는 비정상 입력을 전송하였는데, 비정상 입력의 전송에도 불구하고 정상 응답을 수신하는 경우 제 3 상태(s_2)로 전이된 것으로 보고, 이후에 제 3 상태(s_2)에 맞는 비정상 입력을 전송한다.

[0031] (b)에 도시된 바와 같이, 제 2 상태(s_1)의 상태 머신에 대하여 해당 상태에 맞는 비정상 입력을 전송하였는데, 비정상 입력을 거부하는 취지로 실패응답을 수신한 경우, 실패 응답이 나타내는 상기 유한 상태 머신의 변경 상태에 따라 최초 상태로 상태 전이가 이루어진 것으로 보고, 해당 상태에 맞는 비정상 입력을 전송한다. 비정상 입력에 대하여 실패 응답을 출력하는 경우 해당 상태 머신은 정상적으로 응답한 것으로 볼 수 있다.

[0032] (c)에 도시된 바와 같이, 제 2 상태(s_1)의 상태 머신에 대하여 해당 상태에 맞는 비정상 입력을 전송하였는데, 제한 시간내에 응답을 수신하지 못하는 경우가 발생할 수 있으며, 이에 대해서는 리셋 처리를 수행하며 구체적인 처리 방법에 대해서는 추후 설명하기로 한다.

[0033] 도 5는 종래의 퍼징 방법을 나타내는 유사 알고리즘을 도시한 도면이다.

[0034] 종래의 방법에서는 유한 상태 머신의 각각의 상태별로 비정상 입력을 전송하고, 상태 전이가 발생하는지 여부를 확인한다. 이때, 모든 상태를 점검하는 방식(stateful fuzzing)에 따라 유한 상태 머신의 모든 상태를 고려하여 취약점을 점검한다. 그리고, 비정상 입력에 대하여 정상 응답이 발생할지, 비정상 응답이 발생할지에 대해

서는 예측하기 어려우므로, 해당 방법에 대해서는 응답의 정상 여부와 무관하게, 연결을 리셋하고 모든 상태를 점검하는 방식을 사용한다.

[0035] 본 발명에서는 이러한 종래 기술을 개선하기 위하여, 상태 별로 비정상 입력을 전송하고 이에 대하여 응답이 수신되면, 리셋 단계를 수행하지 않고 다음 단계로 상태를 전이하여, 퍼징을 수행하는 방법으로 알고리즘 최적화를 수행한다.

[0036] 도 6은 본 발명의 일 실시예에 따른 퍼징 방법을 나타내는 유사 알고리즘을 도시한 도면이고, 도 8은 본 발명의 일 실시예에 따른 퍼징 방법을 도시한 순서도이다. 또한, 도 7은 본 발명의 일 실시예에 따른 퍼징 방법에서 비정상 입력을 전송하는 순서를 정하는 방법을 나타내는 유사 알고리즘을 도시한 도면이다.

[0037] 먼저, 도 8에 도시된 바와 같이, 취약점 확인 대상 장치의 유한 상태 머신에 대하여, 제 n 상태(n은 자연수)의 유한 상태 머신에 대하여 해당 상태에 맞게 설정된 비정상 입력을 전송한다(S810). 이때, 도 6의 3번째 라인을 수행함에 따라 도 7의 popInput 모듈을 수행한다. 해당 모듈은 비정상 입력을 전송하는 과정에서 유한 상태 머신의 현재 상태를 참조해 최적의 순서에 따라 비정상 입력을 전송하여, 최소한의 리소스가 사용되도록 한다.

[0038] 도 4를 참조하여 설명하면,s_0 에서 s_1으로 상태 변화를 발생시키는 입력을 i_{01},s_1에서 s_2로 상태 변화를 발생시키는 입력을 i_{12}라고 하고, 각 입력(i_{01}과 i_{12})에 대해 비정상인 입력을 2번씩 보낸다고 가정한다. 이때, 비정상 입력을 전송하는 순서는 i_{01}, i_{01}, i_{12}, i_{12}의 순서로 전송하는 방식과, i_{12}, i_{12}, i_{01}, i_{01} 의 순서로 전송하는 방식,i_{01}, i_{12}, i_{01}, i_{12} 의 순서로 전송하는 방식등 여러가지를 고려할 수 있다.

[0039] 이때, 두번째 방식에 따라 상태 점검을 진행하는 경우, i_{12}를 모두 점검하고 i_{01}를 점검할 때, 비정상 입력(i_{01})을 전송했지만, 유한 상태 머신이 s_1으로 상태 전이가 일어나는 경우, 해당 상태에서는 더 점검할 입력이 없게 된다. 앞선 순서에서 i_{12}를 모두 점검하였기 때문이다. 이에, 다시 s_0으로 상태를 이동시키는데, 이는 리셋 처리를 수행한 것과 동일하게 된다.

[0040] 따라서, 도 4의 경우 비정상 입력(i_{01})이 정상 응답을 유발할 경우 i_{01}, i_{12}, i_{01}, i_{12} 의 순서로 전송하는 방식에 따르면, 가장 최적의 점검을 수행할 수 있다. 즉, 주기적으로 현재 상태에서 가장 짧은 거리에 위치한 상태로 이동하도록 경로를 탐색하고, 이를 기준으로 비정상 입력을 전송한다. 한편, 이때 거리(distance)는 Floyd-Warshall 알고리즘 등을 이용하여 산출할 수 있다.

[0041] 다음으로, 이러한 비정상 입력에 대하여 정상 응답 또는 실패 응답을 수신하는 경우, 다음 상태로 상태가 전이된 것으로 판단하고, 해당 응답이 나타내는 변경 상태에 맞는 비정상 입력을 전송한다(S820, S830). 이에 대해서는 도 6의 3 내지 10 라인에 나타난 알고리즘을 통해 수행된다.

[0042] 그리고, 확인 대상 장치가 해당 비정상 입력에 대하여 취약하지 않음을 기록한다.

[0043] 만약, 제 n 상태의 유한 상태 머신으로부터 제한 시간내에 응답을 수신하지 못하는 경우 리셋 처리를 통해 최초 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행한다(S820).

[0044] 보다 구체적으로 살펴보면, 비정상 입력에 대하여 응답을 제한 시간 내에 수신하지 못한 경우, 제 n 상태의 유한 상태 머신에 대하여 제 n 상태에 맞는 정상 입력을 전송한다(S840). 비정상 입력에 대하여 유한 상태 머신이 비정상 입력을 무시(drop)하거나 예측하지 못한 다른 상태로 전이되었다고 가정할 수 있으므로, 이를 확인하기 위하여 정상 입력을 전송하며, 이에 대하여 정상 응답이 수신되면(S850), 리셋 처리를 수행하지 않고, 유한 상태 머신이 정상 동작하고 있는 것으로 판단하고, 변경 상태에 맞는 비정상 입력을 전송한다(S830).

[0045] 그러나, 정상 입력에 대하여 제 n 상태의 유한 상태 머신으로부터 응답을 수신하지 못한 경우, 리셋 처리를 통해 초기 상태의 유한 상태 머신에 대하여 비정상 입력을 전송하는 절차를 수행한다(S860). 이에 대해서는 도 6의 12 내지 18 라인에 나타난 알고리즘을 통해 수행된다.

[0046] 한편, 앞서 도 3을 통해 설명한 바와 같이, 취약점 확인 대상 장치의 유한 상태 머신에 대하여 각 상태별로 상태 전이를 위해 입력을 전송할 때 응답이 수신되는데 소요되는 평균 시간을 측정하고, 평균 시간에 기초하여 상태 전이 별로 가중치를 설정한다. 그리고, 앞선 단계(S820)에서 기준이 되는 제한 시간은 각 상태 전이별 가중치에 기초하여 설정되도록 한다.

[0047] 이상에서 설명한 본 발명의 실시예에 따른 네트워크 프로토콜의 취약점을 탐지하는 퍼징 방법은, 컴퓨터에 의해

실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행 가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 이러한 기록 매체는 컴퓨터 판독 가능 매체를 포함하며, 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스 될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체를 포함하며, 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다.

[0048] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0049] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0050] 100: 퍼징 장치

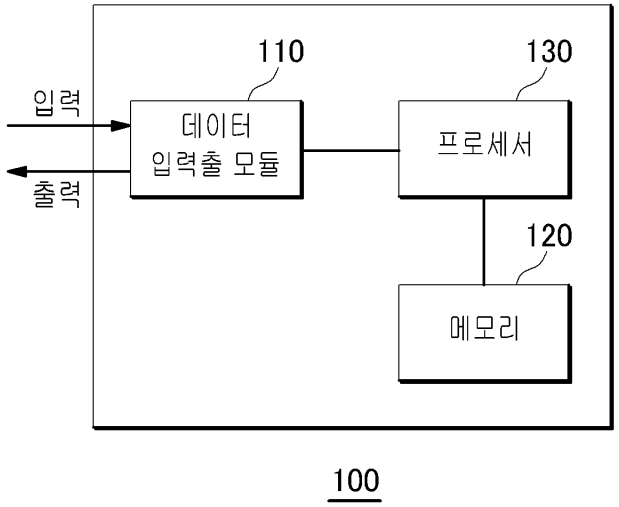
110: 데이터 입출력 모듈

120: 메모리

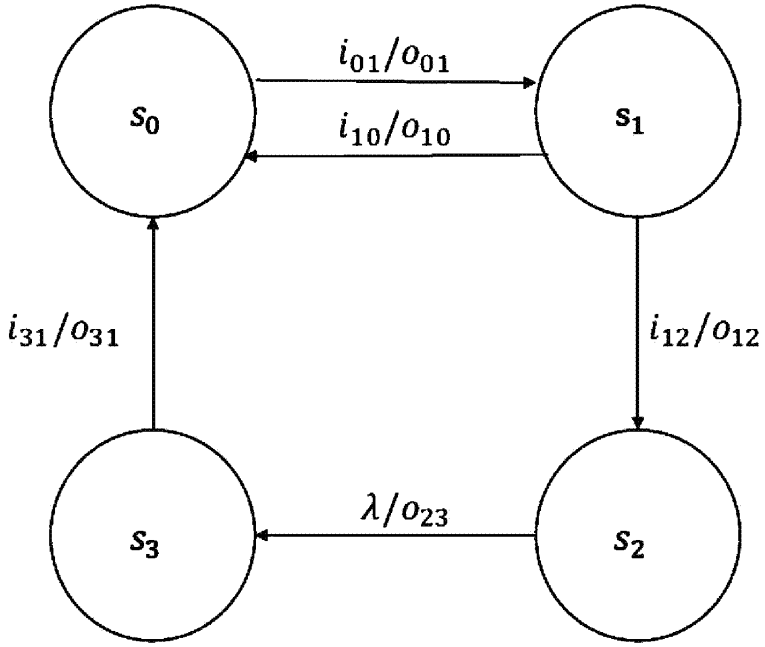
130: 프로세서

도면

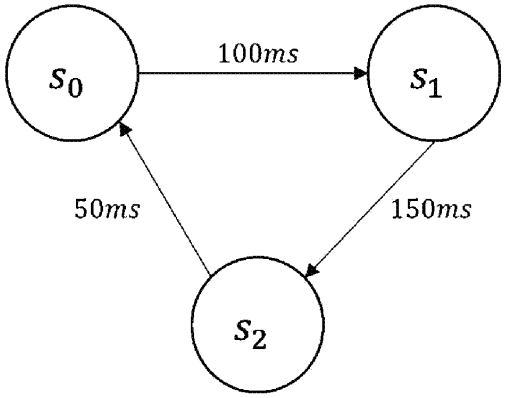
도면1



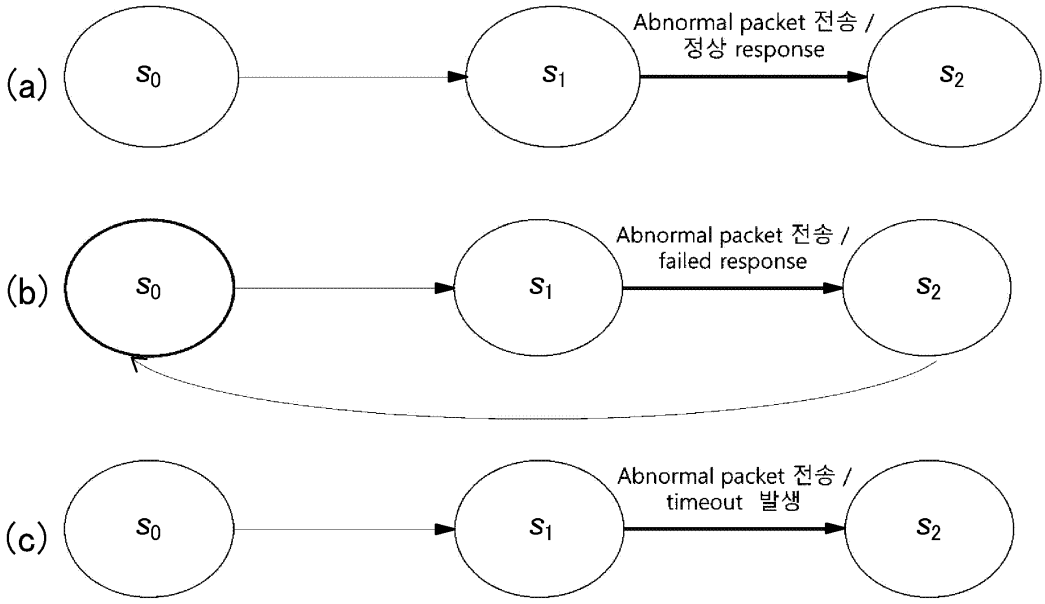
도면2


s_0
s_1
s_2
s_3
i_{01}/o_{01}
i_{10}/o_{10}
i_{12}/o_{12}
i_{31}/o_{31}
λ/o_{23}

도면3


s_0
s_1
s_2
100ms
50ms
150ms

도면4



도면5

Algorithm 1 Reset Algorithm

Input: $M = (S, s_0, I, O, D, \delta, \gamma, \omega)$, D_m: abnormal input domain

1: **for** $\forall(i_m, s_x) \in D_m$ **do**
2: Transit to the state s_x
3: Send i_m to the target
4: Reset the connection
5: Check the crash
6: **end for**

도면6

Algorithm 2 Response Based Algorithm

Input: $M = (S, s_0, I, O, D, \delta, \gamma, \omega)$, D_m: abnormal input domain

1: $cur = s_0$ ▷ Set current state to the initial state
2: **while** $D_m \neq \varnothing$ **do**
3: $(s_x, i_m) = popInput(D_m, cur, M)$ ▷ Algorithm 3
4: Transit to the state s_x
5: Send i_m to the target
6: **if** $\gamma(s_x, i_m^{-1}) \neq \lambda$ **then**
7: $resp = recv()$ ▷ Get response from the target
8: **if** $\exists (s_x, i) \in D \ \ s.t. \ \ \gamma(s_x, i) = resp$ **then**
9: $cur = \delta(s_x, i)$
10: **go to** 3
11: **end if**
12: send i_m^{-1} to the target ▷ Timeout occurred
13: **if** transition success **then**
14: $cur = \delta(s_x, i_m^{-1})$
15: **go to** 3
16: **end if**
17: **end if**
18: Reset the connection
19: $cur = s_0$
20: Check the crash
21: **end while**

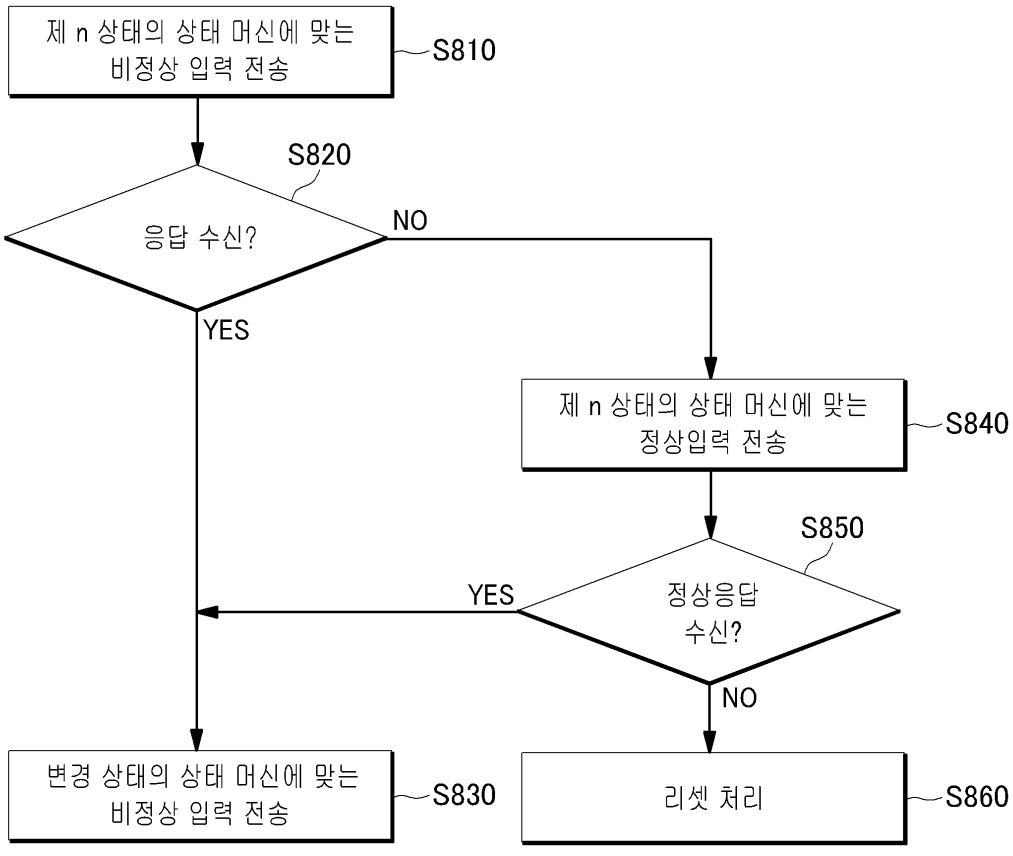
도면7

Algorithm 3 popInput

Input: $M = (S, s_0, I, O, D, \delta, \gamma, \omega)$, D_m: abnormal input domain, cur: current state

$min = \infty$
$input = 0$
for $\forall (s_x, i_m) \in D_m$ **do**
    $distance = CalculateDistance(cur, s_x)$
    if $distance < min$ **then**
        $input = (s_x, i_m)$
        $min = distance$
    end if
end for
$D_m = D_m \setminus input$
return input

도면8



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 7

【변경전】

전송하는고,

【변경후】

전송하는,