(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년03월29일
(11) 등록번호 10-1963756
(24) 등록일자 2019년03월25일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) *G06F 21/56* (2013.01)
G06N 3/08 (2006.01)
(52) CPC특허분류
G06F 21/577 (2013.01)
G06F 11/3604 (2013.01)
(21) 출원번호 10-2018-0142533
(22) 출원일자 2018년11월19일
심사청구일자 2018년11월19일
(56) 선행기술조사문헌
KR1020170003356 A*
KR1020180060497 A*
US10114954 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
윤주범
서울특별시 송파구 충민로4길 19, 704동 401호(장지동, 송파파인타운7단지)
최민준
인천광역시 부평구 세월천로 16, 107동 2504호(청천동, 청천푸르지오아파트)
(74) 대리인
두호특허법인

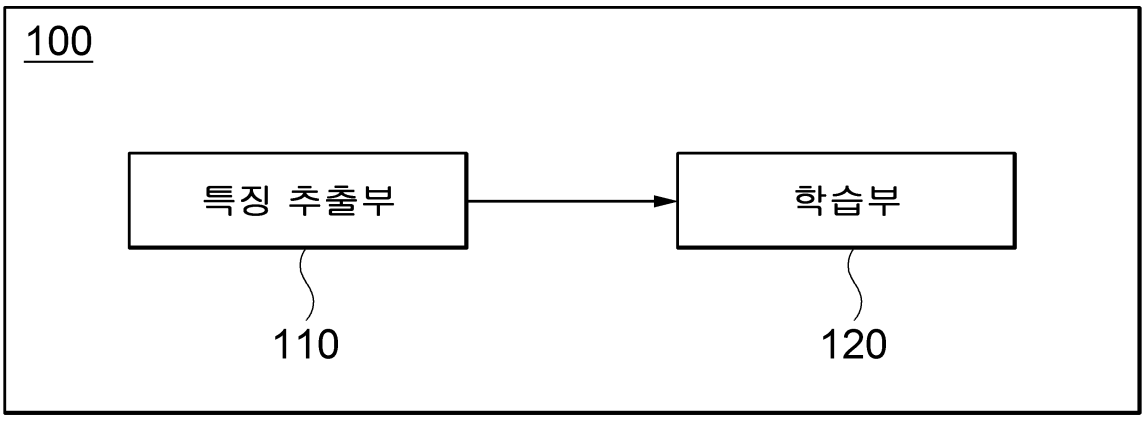
전체 청구항 수 : 총 16 항 심사관 : 구대성

(54) 발명의 명칭 **소프트웨어 취약점 예측 모델 학습 장치 및 방법, 소프트웨어 취약점 분석 장치 및 방법**

(57) 요 약

본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치는, 사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출하는 특징 추출부; 및 상기 복수의 바이너리 파일 각각에 대한 상기 특징 정보 및 취약 여부를 학습 데이터로 이용하여 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 학습시키는 학습부를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 21/563 (2013.01)
G06N 3/08 (2013.01)
G06F 2221/033 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711075702
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 정보통신기술인력양성
연구과제명 지능형 비행로봇 융합기술 연구
기 여 율 1/1
주관기관 세종대학교산학협력단
연구기간 2018.06.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출하는 특징 추출부; 및

상기 복수의 바이너리 파일 각각에 대한 상기 특징 정보 및 취약 여부를 학습 데이터로 이용하여 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 학습시키는 학습부를 포함하며,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함하는 소프트웨어 취약점 예측 모델 학습 장치.

청구항 2

삭제

청구항 3

청구항 1에 있어서,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함하는 소프트웨어 취약점 예측 모델 학습 장치.

청구항 4

청구항 1에 있어서,

상기 복수의 소프트웨어 바이너리 각각에 대한 취약 여부를 분류하는 분류부를 더 포함하고,

상기 학습부는, 상기 특징 정보 및 상기 취약 여부 분류 결과를 학습 데이터로 이용하여 상기 취약점 예측 모델을 학습시키는 소프트웨어 취약점 예측 모델 학습 장치.

청구항 5

청구항 1에 있어서,

상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델인 소프트웨어 취약점 예측 모델 학습 장치.

청구항 6

사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출하는 단계; 및

상기 복수의 바이너리 파일 각각에 대한 상기 특징 정보 및 취약 여부를 학습 데이터로 이용하여 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 학습시키는 단계를 포함하며,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함하는 소프트웨어 취약점 예측 모델 학습 방법.

청구항 7

삭제

청구항 8

청구항 6에 있어서,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함하는 소프트웨어 취약점 예측 모델 학습 방법.

청구항 9

청구항 6에 있어서,

상기 복수의 소프트웨어 바이너리 각각에 대한 취약 여부를 분류하는 단계를 더 포함하고,

상기 학습시키는 단계는, 상기 특징 정보 및 상기 취약 여부 분류 결과를 학습 데이터로 이용하여 상기 취약점 예측 모델을 학습시키는 소프트웨어 취약점 예측 모델 학습 방법.

청구항 10

청구항 6에 있어서,

상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델인 소프트웨어 취약점 예측 모델 학습 방법.

청구항 11

취약점 분석 대상인 소프트웨어 바이너리 파일에 대한 특징 정보를 추출하는 특징 추출부; 및

기 학습된 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 이용하여 상기 추출된 특징 정보로부터 상기 소프트웨어 바이너리 파일의 취약 여부를 판단하는 취약점 판단부를 포함하며,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함하는 소프트웨어 취약점 분석 장치.

청구항 12

삭제

청구항 13

청구항 11에 있어서,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함하는 소프트웨어 취약점 분석 장치.

청구항 14

청구항 11에 있어서,

상기 취약점 예측 모델은, 사전 수집된 복수의 소프트웨어 바이너리 각각에 대한 특징 정보 및 취약 여부를 학습 데이터로 이용하여 사전 학습되는 소프트웨어 취약점 분석 장치.

청구항 15

청구항 11에 있어서,

상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델인 소프트웨어 취약점 분석 장치.

청구항 16

취약점 분석 대상인 소프트웨어 바이너리 파일에 대한 특징 정보를 추출하는 단계; 및

기 학습된 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 이용하여 상기 추출된 특징 정보로부터 상기 소프트웨어 바이너리 파일의 취약 여부를 판단하는 단계를 포함하며,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함하는 소프트웨어 취약점 분석 방법.

청구항 17

삭제

청구항 18

청구항 16에 있어서,

상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함하는 소프트웨어 취약점 분석 방법.

청구항 19

청구항 16에 있어서,

상기 취약점 예측 모델은, 사전 수집된 복수의 소프트웨어 바이너리 각각에 대한 특징 정보 및 취약 여부를 학습 데이터로 이용하여 사전 학습되는 소프트웨어 취약점 분석 방법.

청구항 20

청구항 16에 있어서,

상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델인 소프트웨어 취약점 분석 방법.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시예들은 소프트웨어 취약점 분석 기술과 관련된다.

배 경 기 술

[0002] 4차 산업혁명 시대를 맞아 인공지능, 빅데이터, IoT, 클라우드 등과 같이 다양한 분야를 위한 많은 소프트웨어가 새롭게 개발되고 있다. 하지만, 소프트웨어의 증가에 따라 필연적으로 소프트웨어 취약점도 증가하고 있다. 이러한 소프트웨어 취약점을 통해 해커들이 시스템을 공격할 경우 개인이나 기업은 막대한 피해를 받을 수 있기 때문에 소프트웨어 취약점은 사전에 탐지하여 제거해야 한다.

[0003] 현재 소프트웨어 취약점 검사를 위한 대표적 기술로는 퍼징(fuzzing) 기법과 심볼릭 익스큐션(symbolic execution)을 사용한 기술들이 있다. 하지만, 이러한 종래 기술들은 복잡하고 시간이 오래 걸리는 치명적인 단점이 있다. 이러한 단점을 극복하기 위해 기계학습과 신경망 알고리즘을 사용하여 취약 여부를 예측하는 연구들이 있지만, 예측 정확도가 높지 않거나 소프트웨어의 소스 코드가 필요하다는 한계점이 존재한다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 대한민국 등록특허 제10-1640479호 (2018.07.18. 공고)

발명의 내용

해결하려는 과제

[0005] 본 발명의 실시예들은 소프트웨어 취약점 예측 모델 학습 장치 및 방법과 소프트웨어 취약점 분석 장치 및 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0006] 본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치는, 사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출하는 특징 추출부; 및 상기 복수의 바이너리 파일 각각에 대한 상기 특징 정보 및 취약 여부를 학습 데이터로 이용하여 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 학습시키는 학습부를 포함한다.

[0007] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다.

[0008] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0009] 상기 복수의 소프트웨어 바이너리 각각에 대한 취약 여부를 분류하는 분류부를 더 포함하고, 상기 학습부는, 상기 특징 정보 및 상기 취약 여부 분류 결과를 학습 데이터로 이용하여 상기 취약점 예측 모델을 학습시킬 수 있다.

[0010] 상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델일 수 있다.

[0011] 본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 방법은, 사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출하는 단계; 및 상기 복수의 바이너리 파일 각각에 대한 상기 특징 정보 및 취약 여부를 학습 데이터로 이용하여 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 학습시키는 단계를 포함한다.

[0012] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다.

[0013] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0014] 상기 복수의 소프트웨어 바이너리 각각에 대한 취약 여부를 분류하는 단계를 더 포함하고, 상기 학습시키는 단계는, 상기 특징 정보 및 상기 취약 여부 분류 결과를 학습 데이터로 이용하여 상기 취약점 예측 모델을 학습시킬 수 있다.

[0015] 상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델일 수 있다.

[0016] 본 발명의 일 실시예에 따른 소프트웨어 취약점 분석 장치는 취약점 분석 대상인 소프트웨어 바이너리 파일에 대한 특징 정보를 추출하는 특징 추출부; 및 기 학습된 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 이용하여 상기 추출된 특징 정보로부터 상기 소프트웨어 바이너리 파일의 취약 여부를 판단하는 취약점 판단부를 포함한다.

[0017] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다.

[0018] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0019] 상기 취약점 예측 모델은, 사전 수집된 복수의 소프트웨어 바이너리 각각에 대한 특징 정보 및 취약 여부를 학습 데이터로 이용하여 사전 학습될 수 있다.

[0020] 상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델일 수 있다.

[0021] 본 발명의 일 실시예에 따른 소프트웨어 취약점 분석 방법은, 취약점 분석 대상인 소프트웨어 바이너리 파일에 대한 특징 정보를 추출하는 단계; 및 기 학습된 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 이용하여 상기 추출된 특징 정보로부터 상기 소프트웨어 바이너리 파일의 취약 여부를 판단하는 단계를 포함한다.

[0022] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다.

[0023] 상기 특징 정보는, 상기 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 상기 호출되는 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0024] 상기 취약점 예측 모델은, 사전 수집된 복수의 소프트웨어 바이너리 각각에 대한 특징 정보 및 취약 여부를 학습 데이터로 이용하여 사전 학습될 수 있다.

[0025] 상기 취약점 예측 모델은, 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델일 수 있다.

발명의 효과

[0026] 본 발명의 실시예들에 따르면, 사전 수집된 소프트웨어 바이너리 파일로부터 추출된 특징 정보 및 해당 소프트웨어 바이너리 파일에 대한 취약 여부를 이용하여 인공 신경망 기반의 취약점 예측 모델을 학습시키고, 학습된 취약점 예측 모델을 이용하여 임의의 소프트웨어 바이너리 파일에 대한 취약점을 분석함으로써, 소프트웨어의 소스 코드 없이도 정확하고 신속한 취약점 분석이 가능하게 된다.

도면의 간단한 설명

[0027] 도 1은 본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치의 구성도

도 2는 본 발명의 추가적인 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치의 구성도

도 3은 본 발명의 일 실시예에 따른 소프트웨어 취약점 분석 장치의 구성도

도 4는 본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 방법의 순서도

도 5는 본 발명의 추가적인 실시예에 따른 취약점 예측 모델 학습 방법의 순서도

도 6은 본 발명의 일 실시예에 따른 소프트웨어 취약점 분석 방법의 순서도

도 7은 예시적인 실시예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

발명을 실시하기 위한 구체적인 내용

[0028] 이하, 도면을 참조하여 본 발명의 구체적인 실시형태를 설명하기로 한다. 이하의 상세한 설명은 본 명세서에서 기술된 방법, 장치 및/또는 시스템에 대한 포괄적인 이해를 돕기 위해 제공된다. 그러나 이는 예시에 불과하며 본 발명은 이에 제한되지 않는다.

[0029] 본 발명의 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 본 발명의 실시예들을 기술하기 위한 것이며, 결코 제한적이어서는 안 된다. 명확하게 달리 사

용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 특성들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 특성, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0030] 도 1은 본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치의 구성도이다.

[0031] 도 1을 참조하면, 본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치(100)는 특징 추출부(110) 및 학습부(120)를 포함한다.

[0032] 특징 추출부(110)는 사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출한다.

[0033] 이때, 본 발명의 일 실시예에 따르면, 특징 정보는 각 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다. 구체적인 예로, 특징 정보는 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 호출되는 각 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0034] 한편, 본 발명의 일 실시예에 따르면, 특징 정보는 예를 들어, ptrace를 이용하여 추출될 수 있으나, 반드시 이에 한정되는 것은 아니며, 프로세스의 실행 흐름을 추적할 수 있는 공지된 다양한 방법에 의해 추출될 수 있다.

[0035] 학습부(120)는 각 소프트웨어 바이너리 파일에서 추출된 특징 정보 및 각 소프트웨어 바이너리 파일의 취약 여부를 학습 데이터로 이용하여 인공 신경망(artificial neural network) 기반의 취약점 예측 모델을 학습시킨다.

[0036] 구체적으로, 학습부(120)는 수집된 특정 소프트웨어 바이너리 파일에서 추출된 특징 정보에 기초하여 생성된 특징 벡터를 입력 값으로 이용하고, 해당 소프트웨어 바이너리 파일의 취약 여부에 대한 분류 값을 목표 값으로 이용한 지도 학습(supervised learning)을 이용하여 취약점 예측 모델을 학습시킬 수 있다. 이를 통해, 취약점 예측 모델은 임의의 소프트웨어 바이너리 파일의 특징 벡터를 입력하였을 때, 해당 소프트웨어 바이너리 파일의 취약 여부에 대한 분류 값을 출력하도록 학습될 수 있다.

[0037] 한편, 학습부(120)는 실시예에 따라 특징 추출부(110)에 의해 추출된 소프트웨어 바이너리 파일의 특징 정보로부터 특징 벡터를 생성하기 위해 예를 들어, bag-of-words 알고리즘, word2vec 알고리즘 등을 이용한 전처리 과정을 수행할 수 있다.

[0038] 또한, 사전 수집된 소프트웨어 바이너리 파일 각각에 대한 취약 여부는 해당 소프트웨어 바이너리 파일에 대해 사전에 알려진 취약점 정보 등을 이용하여 사용자에 의해 미리 설정될 수 있다.

[0039] 한편, 본 발명의 일 실시예에 따르면, 취약점 예측 모델은 다층 퍼셉트론(Multi-Layer Perceptron) 기반의 인공 신경망 모델일 수 있다. 이때, 다층 퍼셉트론은 단층 퍼셉트론으로 해결할 수 없는 XOR 문제를 해결하기 위한 퍼셉트론으로서, 입력 층과 출력 층 사이에 은닉 층(hidden layer)을 구비한 인공 신경망 모델을 의미한다.

[0040] 도 2는 본 발명의 추가적인 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치의 구성도이다.

[0041] 도 2를 참조하면, 본 발명의 추가적인 실시예에 따른 소프트웨어 취약점 예측 모델 학습 장치(100)는 분류부(130)를 더 포함할 수 있다.

[0042] 구체적으로, 분류부(130)는 취약점 예측 모델 학습을 위해 사전 수집된 소프트웨어 바이너리 파일 각각에 대한 취약 여부를 분류할 수 있다.

[0043] 구체적으로, 분류부(130)는 예를 들어, zzuf 퍼저와 같이 유효하면서 예상치 않은(unexpected) 임의의 데이터를 입력하여 소프트웨어의 취약점 존재 가능성을 판단하기 위한 다양한 방식의 퍼징(fuzzing) 기법을 이용하여 각 소프트웨어 바이너리 파일에 대한 취약 여부를 분류할 수 있다.

[0044] 이 경우, 학습부(120)는 특징 추출부(110)에 의해 추출된 각 소프트웨어 바이너리 파일의 특징 정보에 기초하여 생성된 특징 벡터를 입력 값으로 이용하고, 분류부(130)에 의해 결정된 각 소프트웨어 바이너리 파일의 취약 여부에 대한 분류 결과를 목표 값으로 이용한 지도 학습을 이용하여 취약점 예측 모델을 학습시킬 수 있다.

[0045] 도 3은 본 발명의 일 실시예에 따른 소프트웨어 취약점 분석 장치의 구성도이다.

[0046] 도 3을 참조하면, 본 발명의 일 실시예에 따른 소프트웨어 취약점 분석 장치(300)는 특징 추출부(310) 및 취약점 판단부(320)를 포함한다.

[0047] 특징 추출부(310)는 취약점 분석 대상인 소프트웨어 바이너리 파일에 대한 특징 정보를 추출한다.

[0048] 이때, 특징 추출부(310)에 의해 추출되는 특징 정보 및 특징 정보 추출 방식은 도 1에 도시된 특징 추출부(110)와 동일하므로, 이에 대한 중복적인 설명은 생략한다.

[0049] 취약점 판단부(320)는 기 학습된 인공 신경망 기반의 취약점 예측 모델을 이용하여 특징 추출부(310)에 의해 추출된 특징 정보로부터 취약점 분석 대상인 소프트웨어 바이너리 파일의 취약 여부를 판단한다.

[0050] 이때, 취약점 예측 모델은 예를 들어, 도 1 또는 도 2에 도시된 취약점 예측 모델 학습 장치(100)에 의해 사전 학습될 수 있다.

[0051] 구체적으로, 취약점 판단(320)는 취약점 판단 대상인 소프트웨어 바이너리 파일에서 추출된 특징 정보에 기초하여 생성된 특징 벡터를 취약점 예측 모델로 입력하고, 취약점 예측 모델로부터 출력되는 취약 여부에 대한 분류 결과에 기초하여 취약점 판단 대상인 소프트웨어 바이너리 파일의 취약 여부를 판단할 수 있다.

[0052] 한편, 취약점 판단부(320)는 실시예에 따라 특징 추출부(310)에 의해 추출된 특정 바이너리 파일의 특징 정보로부터 특징 벡터를 생성하기 위해 예를 들어, bag-of-words 알고리즘, word2vec 알고리즘 등을 이용한 전처리 과정을 수행할 수 있다.

[0053] 도 4는 본 발명의 일 실시예에 따른 소프트웨어 취약점 예측 모델 학습 방법의 순서도이다.

[0054] 도 4에 도시된 방법은 예를 들어, 도 1에 도시된 소프트웨어 취약점 예측 모델 학습 장치(100)에 의해 수행될 수 있다.

[0055] 도 4를 참조하면, 우선, 소프트웨어 취약점 예측 모델 학습 장치(100)는 사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출한다(410).

[0056] 이때, 본 발명의 일 실시예에 따르면, 특징 정보는 각 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다. 구체적인 예로, 특징 정보는 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 호출되는 각 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0057] 이후, 소프트웨어 취약점 예측 모델 학습 장치(100)는 각 소프트웨어 바이너리 파일에 대한 특징 정보 및 취약 여부를 학습 데이터로 이용하여 인공 신경망기반의 취약점 예측 모델을 학습시킨다(420).

[0058] 이때, 각 소프트웨어 바이너리 파일의 취약 여부는 사용자에 의해 미리 설정될 수 있다.

[0059] 도 5는 본 발명의 추가적인 실시예에 따른 취약점 예측 모델 학습 방법의 순서도이다.

[0060] 도 5에 도시된 방법은 예를 들어, 도 2에 도시된 소프트웨어 취약점 예측 모델 학습 장치(100)에 의해 수행될 수 있다.

[0061] 도 5를 참조하면, 우선, 소프트웨어 취약점 예측 모델 학습 장치(100)는 사전 수집된 복수의 소프트웨어 바이너리 파일 각각에 대한 특징 정보를 추출한다(510).

[0062] 이때, 본 발명의 일 실시예에 따르면, 특징 정보는 각 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다. 구체적인 예로, 특징 정보는 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 호출되는 각 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0063] 이후, 소프트웨어 취약점 예측 모델 학습 장치(100)는 복수의 소프트웨어 바이너리 파일 각각에 대한 취약 여부를 분류할 수 있다(520).

[0064] 이후, 소프트웨어 취약점 예측 모델 학습 장치(100)는 각 소프트웨어 바이너리 파일에 대한 특징 정보 및 취약 여부 분류 결과를 학습 데이터로 이용하여 인공 신경망 기반의 취약점 예측 모델을 학습시킨다(530).

[0065] 도 6은 본 발명의 일 실시예에 따른 소프트웨어 취약점 분석 방법의 순서도이다.

[0066] 도 6에 도시된 방법은 예를 들어, 도 3에 도시된 소프트웨어 취약점 분석 장치(300)에 의해 수행될 수 있다.

[0067] 도 6을 참조하면, 우선, 소프트웨어 취약점 분석 장치(300)는 취약점 분석 대상인 소프트웨어 바이너리 파일에 대한 특징 정보를 추출한다(610).

[0068] 이때, 본 발명의 일 실시예에 따르면, 특징 정보는 각 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들에 대한 정보를 포함할 수 있다. 구체적인 예로, 특징 정보는 소프트웨어 바이너리 파일이 실행될 때 호출되는 함수들의 리스트 및 호출되는 각 함수들의 호출 순서 중 적어도 하나를 포함할 수 있다.

[0069] 이후, 소프트웨어 취약점 분석 장치(300)는 기 학습된 인공 신경망 기반의 취약점 예측 모델을 이용하여, 추출된 특징 정보로부터 취약점 분석 대상인 소프트웨어 바이너리 파일의 취약 여부를 판단한다(620).

[0070] 이때, 본 발명의 일 실시예에 따르면, 취약점 예측 모델은 예를 들어, 도 4 또는 도 5에 도시된 방법에 의해 사전 학습될 수 있다.

[0071] 한편, 도 4 내지 도 6에 도시된 순서도에서는 상기 방법을 복수 개의 단계로 나누어 기재하였으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다.

[0072] 도 7은 예시적인 실시예들에서 사용되기에 적합한 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도이다. 도시된 실시예에서, 각 컴포넌트들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 이하에 기술되지 않은 것 이외에도 추가적인 컴포넌트를 포함할 수 있다.

[0073] 도시된 컴퓨팅 환경(10)은 컴퓨팅 장치(12)를 포함한다. 일 실시예에서, 컴퓨팅 장치(12)는 취약점 예측 모델 학습 장치(100) 또는 소프트웨어 취약점 분석 장치(300)에 포함되는 하나 이상의 컴포넌트일 수 있다.

[0074] 컴퓨팅 장치(12)는 적어도 하나의 프로세서(14), 컴퓨터 판독 가능 저장 매체(16) 및 통신 버스(18)를 포함한다. 프로세서(14)는 컴퓨팅 장치(12)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로세서(14)는 컴퓨터 판독 가능 저장 매체(16)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 상기 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨터 실행 가능 명령어는 프로세서(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0075] 컴퓨터 판독 가능 저장 매체(16)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능 저장 매체(16)에 저장된 프로그램(20)은 프로세서(14)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독 가능 저장 매체(16)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 컴퓨팅 장치(12)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0076] 통신 버스(18)는 프로세서(14), 컴퓨터 판독 가능 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0077] 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.

[0078] 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 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

[0079] 10: 컴퓨팅 환경

12: 컴퓨팅 장치

14: 프로세서

16: 컴퓨터 판독 가능 저장 매체

18: 통신 버스

20: 프로그램

22: 입출력 인터페이스

24: 입출력 장치

26: 네트워크 통신 인터페이스

100: 소프트웨어 취약점 예측 모델 학습 장치

110: 특징 추출부

120; 학습부

130: 분류부

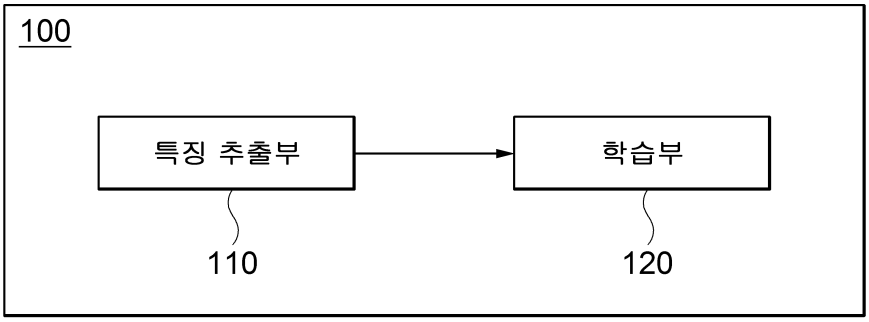
300: 소프트웨어 취약점 분석 장치

310: 특징 추출부

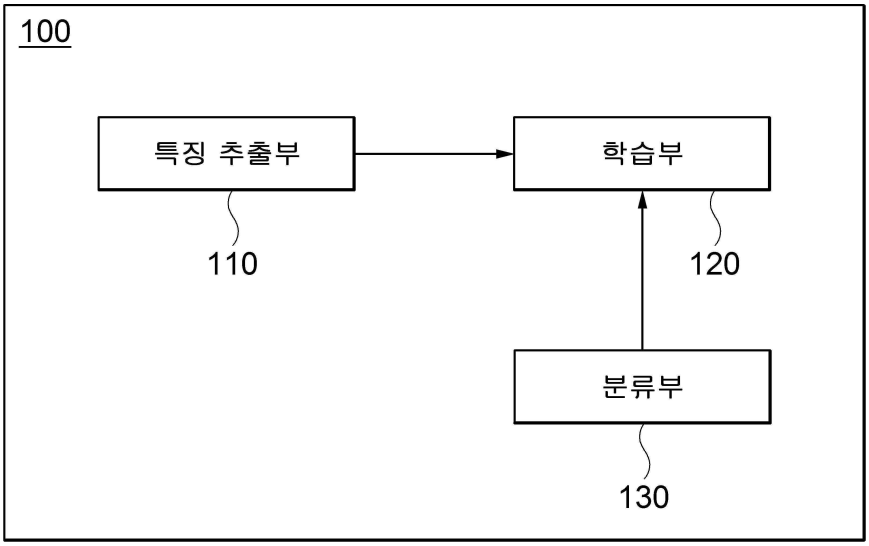
320: 취약점 판단부

도면

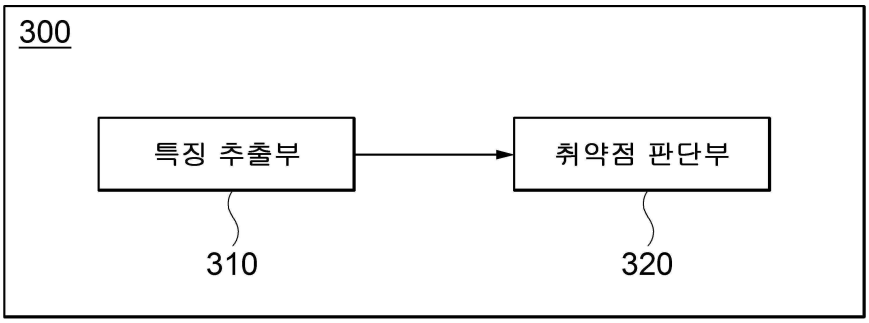
도면1



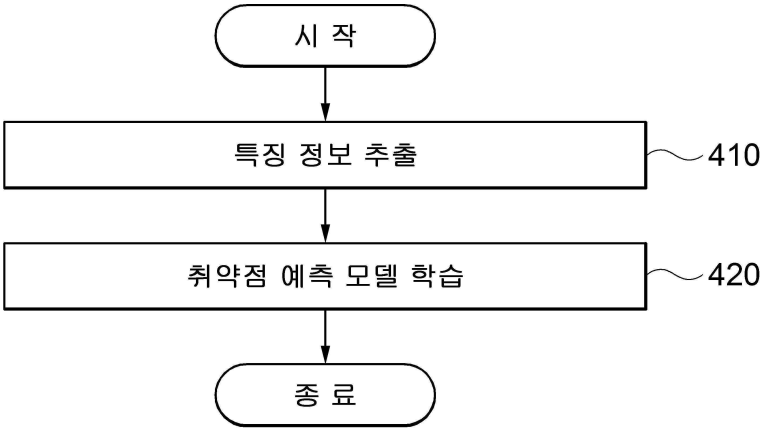
도면2



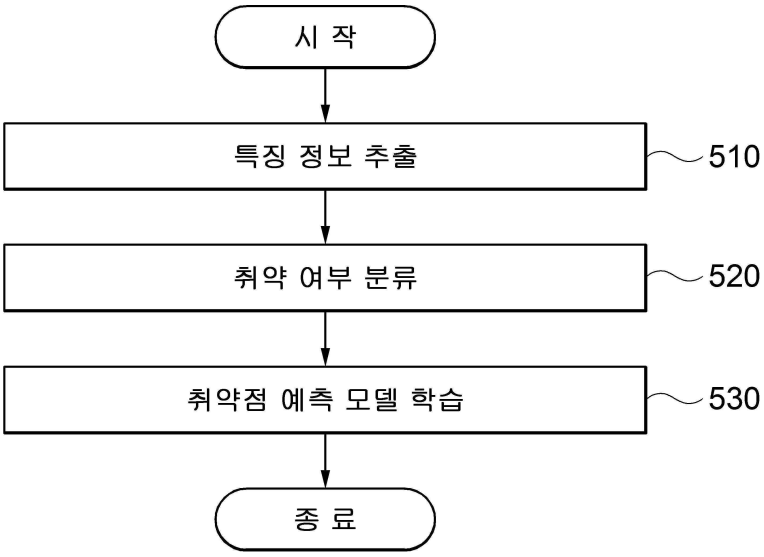
도면3


300
특징 추출부
취약점 판단부
310
320

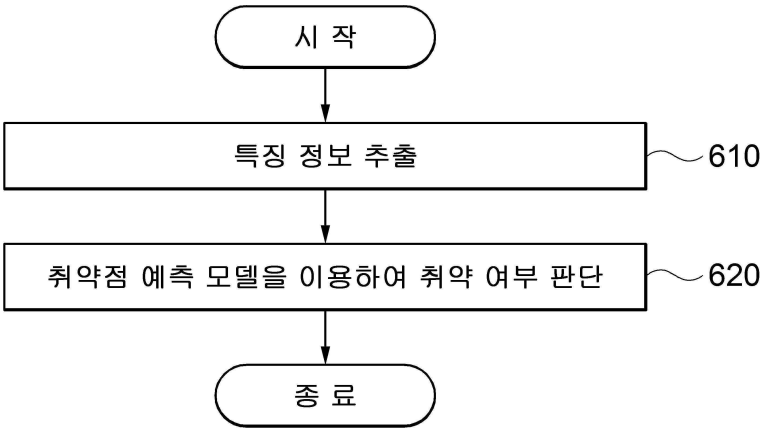
도면4


시 작
특징 정보 추출
410
취약점 예측 모델 학습
420
종 료

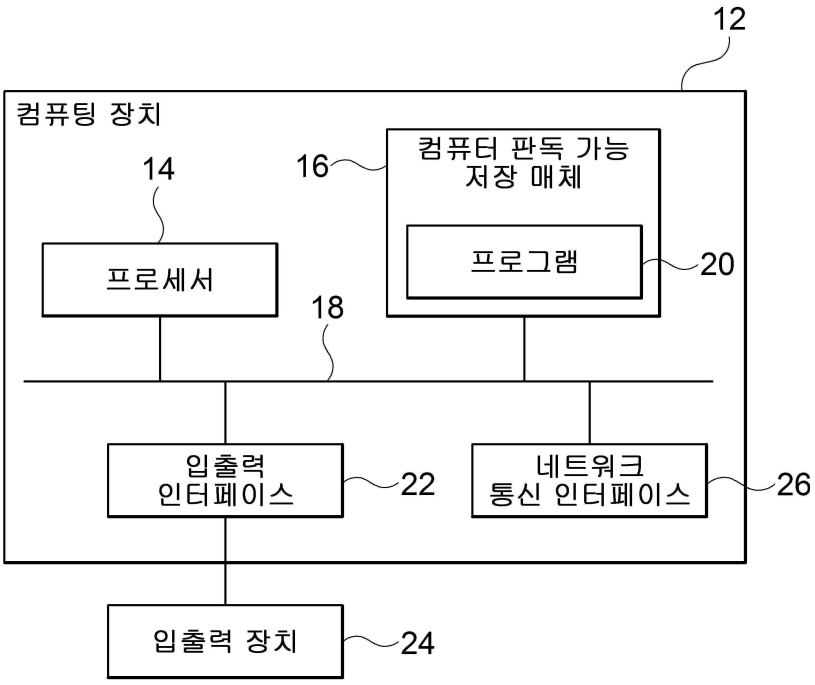
도면5


시 작
특징 정보 추출
510
취약 여부 분류
520
취약점 예측 모델 학습
530
종 료

도면6


시 작
특징 정보 추출
610
취약점 예측 모델을 이용하여 취약 여부 판단
620
종 료

도면7


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