BERT 와 SmartConDataset 을 이용해 스마트 계약의 취약점 분석

이진규
성균관대학교

leejk5152@g.skku.edu

Vulnerability analysis of smart contracts using BERT and SmartConDataset

Lee Jin Kyu
Sungkyunkwan Univ.

요 약

블록체인 기술은 오늘날 암호화폐의 급성장으로 인해 알려지며 차세대 4 차 산업혁명 시대의 핵심적인 보안 기술로 급부상하고 있다. 이미 여러 분야에서 블록체인 기술을 활용해 보안을 강화하려고 연구 중에 있으며, 다양한 증명서 발급에 핵심 기술로 주목받고 있다. 블록체인 기술은 이미 검증된 보안기술로써 쉽게 위조나 해킹하기 어려울 정도의 보안성을 갖추고 있으나, 블록체인이 각광받는 동시에 블록체인의 다양한 취약점이 드러나 보안에 위협이 되고 있다. 본 논문에서는 스마트 계약의 취약성을 탐지하기 위해 사전 훈련된 모델인 BERT 모델을 사용하였다. 본 논문은 SmartCheck[1]의 취약점 기준 43 가지를 이용하여 취약점을 하였고, SmartConDetect[2]의 데이터셋을 활용하여 BERT 모델을 사전 훈련시켰다. 이전 논문과는 달리 본 논문에서는 취약점이 없는 정상 코드를 추가로 훈련시켜 라벨링 없이 정상적인 코드를 구별할 수 있게 하였다. 이를 통해 Ehterscan 에 공개되어 있는 스마트 계약의 취약점을 탐지해 보았다.

Ⅰ. 서 론

블록체인 기술은 오늘날 암호화폐의 급성장으로 인해 알려지며 차세대 4 차 산업혁명 시대의 핵심적인 보안 기술로 급부상하고 있다. 블록체인 기술은 이미 검증된 보안기술로써 쉽게 위조나 해킹하기 어려울 정도의 보안성을 갖추고 있으나, 블록체인이 각광받는 동시에 블록체인의 다양한 취약점이 드러나 보안에 위협이 되고 있다. 기존에는 전문가가 수동으로 직접 코드를 분석하여 취약점을 분석한 후 제거했지만 자연어처리와 기계학습의 발달로 자동으로 취약점을 분석하는 기술이 발전하고 있다.

정적 분석은 코드에서 취약성을 실행하지 않고 찾는 방법이다. 정적 분석 도구는 코드를 실행하지 않고 모든 코드를 검사하여 신속하게 취약점을 탐지할 수 있고 프로그램이 완성되지 않더라도 이전에 취약점을 탐지할 수 있다. 기존의 취약성 분석 도구로는 RNN 모델이 있다. 그러나 RNN 모델은 forward direction 으로 취약점을 탐지하기 때문에 사전데이터의 훈련 순서가 결과에 영향을 미친다.[3] BERT 모델은 이와 같은 RNN 모델의 단점을 없애고 Bi-direction 으로 취약점을 탐지할 수 있는 모델이다. 또한 BERT 모델은 RNN 모델과 달리 다른 단어 그리고 다른 문장과의 관계를 계산하여 취약성을 탐지할 수 있다.

Ⅱ. 본론

본 논문에서는 'SmartConDataSet'[2]을 사전 훈련을 위한 데이터셋으로 활용한다. 'SmartConDataset'은 Eherscan 에서 2021 년 4 월부터 5 월까지 4 주 동안 10000 개의 solidity 파일을 크롤링해서 취약점을 분석한

데이터셋이다. ‘SmartConDataset’에서는 크롤링한 solidity 파일을 ‘Smartcheck’를 통해 취약점 기준 43 가지로 취약점이 있는 코드를 탐지하여 모아둔 데이터셋이다. ‘SmartConDataset’은 10000 개의 solidity 파일 중 3053 개의 solidity 파일에서 30 가지 종류의 58059 개의 취약점을 검출하였다.

우선 취약점 이름으로 라벨링된 데이터들을 숫자로 라벨링하였다. ‘SOLIDITY_ADDRESS_HARDCODED’를 숫자 1 로, 시작해서 검출된 총 30 개의 취약점에 각 숫자를 라벨링하였다. 여기서 검출되지 않은 취약점은 데이터에 없기 때문에 탐지할 수 없고 사전 훈련하는 데 시간을 소모하기 때문에 제외하였다. 또한 중복되는 취약점 데이터도 사전 훈련에 시간을 소모하기 때문에 중복되는 데이터를 제거하여 주었다. 남은 4000 개의 데이터 중 훈련 데이터 비율을 90 퍼센트로 하고 남은 10 퍼센트는 테스트 데이터로 활용하였다. 또한 훈련 데이터 중 20 프로는 검증 데이터로 활용하였다.

BERT 모델이 코드를 읽게 하기 위해서 전처리 과정이 필요하다. BERT 의 입력 형식에 맞게 변환하기 위해 각 코드 가젯의 앞에 ‘[CLS]’ 토큰을 삽입하였고 코드 가젯의 끝에 ‘[SEP]’ 토큰을 삽입하였다. 이후, 토큰 임베딩, 세그먼트 임베딩, 포지션 임베딩을 통해 코드가젯을 BERT 로 토큰화하였다.

다음은 토큰의 예시이다.

['[CLS]', 'only', '##A', '##uth', '##ori', '##zed', '##ret', '##urn', '##s', '[SEP]']

BERT 모델이 코드를 읽게 하기 위해서 전처리 과정이

사전 훈련 시 최대의 문장 길이보다 짧은 코드는 뒤의 벡터에 padding 이 필요하다. 여기서 문장 길이가 최대인 코드의 문장 길이가 너무 길면 나머지 토큰의 padding 해야할 양이 많아지고 벡터의 크기가 과하게 커져 사전 훈련 시간이 길어지기 때문에 89 퍼센트의 코드를 그대로 보존할 수 있도록 최대 문장 길이를 256 으로 하였다.

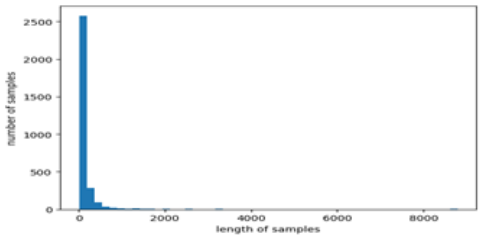

그림 1 코드의 문장 길이

본 논문에서는 위 58059 의 취약점에 취약점이 없는 코드 102 개를 추가하여 BERT 모델에 사전 훈련시켜 취약점을 감지할 수 있도록 하였다.

Ⅲ. 결론

GPU 는 TESLA 4 를 사용하였고, epoch 는 3 으로 사전 훈련 및 테스트를 진행하였다.

그림 2 각 epoch 에서 사전 훈련 및 검증 결과

3 회의 학습 및 검증이 진행되면서 training loss 는 2.17 에서 0.75 로, validation loss 는 1.38 에서 0.73 으로 줄었다. 또한 43 가지의 취약점을 30 가지로 줄이고 취약점이 없는 ‘NO_VUL’라벨을 추가해 31 가지로 분류했기 때문에 5 분 44 초로 시간을 단축할 수 있었다. 모델의 정확도는 SmartConDetect 와 비슷한 86 퍼센트로 나왔다. 그 후 Ehterscan 에서 제공하는 스마트 계약의 코드를 가져와서 훈련된 모델에 적용해 취약점을 검출할 수 있었다..

참 고 문 헌

[1] Tikhomirov S et al., static analysis of ethereum smart contracts, 2018.

[2] S. Jeon et al., Design and evaluation of highly accurate smart contract code vulnerability detection framework, 2023.

[3] Wenhui Jin, Heekuck Oh, A BERT-Based Deep learning Approach for Vulnerability Detection. 2022.