인공신경망 기반 스마트 계약 취약점 탐지 기법 연구 동향

김원웅[1], 엄시우[1], 김현지[1], 강예준[1], 임세진[1], 서화정[1]
[1]한성대학교 IT융합공학과
dnjsdndeee@gmail.com, shuraatum@gamil.com, khj1594012@gmail.com,
etus1211@gmail.com, dlatpwls834@gmail.com, hwajeong84@gmail.com

Trends in Artificial Neural Network based in Smart Contract Vulnernability Detection Technology

Won-Woong Kim[1], Si-Woo Eum[1], Hyun-Ji Kim[1], Yea-Jun Kang[1], Se-Jin Lim[1], Hwa-Jeong Seo[1]
[1]Dept. of IT Convergence Engineering, Hansung University

요 약

최근 데이터의 불변성에 의하여 NFT와 같은 블록체인 기술이 각광받고 있으며, 그에 따라 스마트 계약의 사용량 또한 증가하는 추세이다. 하지만 스마트 계약에 다양한 보안 취약점이 존재하며 배포 후에는 수정을 할 수 없는 스마트 계약 특성상 사전에 해당 취약점을 탐지하는 기술의 중요성이 대두되고 있다. 따라서 본 논문에서는 인공신경망 기반의 스마트 계약의 취약점을 탐지하는 기법 연구 동향에 대하여 알아본다.

1. 서론

현재 데이터를 안전하게 유지할 수 있는 특성에 의해 블록체인 기술이 부상하고 있으며, 이에 따라 스마트 계약의 보안 취약점에 대한 중요성이 부각되고 있다. 스마트 계약의 특성상 거래가 배포된 후에는 수정이 불가능하기에 사용자로부터 신뢰를 얻을 수 있지만, 동시에 잠재적인 보안 문제점을 가지게 된다. 따라서 계약이 이행되기 전에 악의적인 의도를 가진 계약을 탐지하는 것이 매우 중요하다. 또한 스마트 계약은 바이트 코드로써 존재하기 때문에 이를 수동으로 분석하는 것에는 매우 많은 시간적 비용이 필요하다. 따라서 인공신경망을 사용하여 악의적인 스마트 계약의 패턴을 학습하여 탐지하는 것이 매우 효율적이다[2]. [3]에서는 20개의 취약점 패턴을 도메인에 따라 계약 간 취약점, 계약 취약점, 정수 버그, 가스 관련 취약점, 다국적 취약점, 사용되지 않는 취약점, 그리고 무작위 취약점으로 분류하였다. [4]에서는 Solidity, EVM 그리고 Blockchain 단계에서의 스마트 계약 취약성에 대한 분류에 대하여 소개하였다. 또한 재진입 취약점, 이벤트 순서 취약점에 대하여 탐지하는 모델도 소개하였다. 그 외에도 DAO 공격, 패리티 지갑 해킹, ERC-20 캠페인 공격 등이 존재한다[5].

2. 관련 연구

2.1 이더리움

이더리움은 블록체인 플랫폼 중 하나로써, 전용 암호화폐를 이용한 스마트 계약의 실행 및 호출을 처리하기 위하여 EVM(Ethereum Virtual Machine)을 제공한다. 거래가 입력되면 내부적으로 메시지로 변환된 후 실행을 위하여 EVM으로 전달된다. 스마트 계약이 호출되면 EVM의 인터프리터를 통해 바이트 코드가 호출되고 실행된다. 기존 애플리케이션과 달리 스마트 계약의 실행 및 거래는 채굴자에게 계약 실행의 계산 비용 또는 암호화폐로 지불되는 거래에 대한 수수료로 사용되는 단위인 가스를 일정량 보유해야 한다.

2.2 스마트 계약

스마트 계약은 EVM에 의해 컴파일되어 바이트 코드를 생성하고 주소 호출을 통해 블록체인 플랫폼 상에서 실행된다. 스마트 계약은 사전에 설정한 조건이 충족되면 제 3자 없이 두 당사자 간의 거래가 가능해진다. 이러한 특성상 스마트 계약은 거래 과정을 단순화하고 실행 결과 또한 계약 참여자에게

투명하게 공개되므로 블록체인 기반 시스템의 신뢰성을 높이게 된다.

2.2.1 스마트 계약 취약점[5]

스마트 계약은 프로그래밍 언어와 실행 시스템에 결함이 존재한다. 개발자들은 해당 취약점의 정보를 수집하고 SWC(Smart contract Weakness Classfication) 레지스트리를 사용하여 취약점을 분류한다. SWC에는 산술 취약점(SWC-101) 재진입성(SWC-107) 그리고 알려지지 않은 주소가 포함된 거래가 있다[5].

산술 취약점은 오버플로, 언더플로 그리고 산술 문제를 의미한다. 재진입성은 스마트 계약의 기능 중 하나인 외부 계약 코드를 호출하는 기능을 사용하여 외부 사용자 주소로 디지털 통화를 보내 재진입을 유발하는 것을 말한다. 공격자는 재진입 취약성을 사용하여 재귀적 콜백 함수를 수행하여 계약의 계정 잔고가 청산되거나 가스 상한에 도달할 때까지 계약의 출금 작업을 반복하여 수행할 수 있다. 또한 계약에 알 수 없는 주소가 포함되어 있는 경우 해당 주소를 악의적인 활동에 사용할 수 있게 된다[5]. 이 외에도 다양한 취약점이 존재한다.

2.2.2 스마트 계약 취약점 탐지[5]

스마트 계약 취약점 탐지에는 기호 실행, 형식 검증, 퍼지 테스트, ML 기반 방법, 기타 방법과 같이 5가지 유형으로 나눌 수 있다.

기호 실행은 스마트 계약의 소스 코드 또는 바이트 코드에서의 취약점을 탐지하기 위하여 제어 흐름 그래프를 구성하고 다른 도구에 대한 실행 엔진을 제공한다. 형식 검증은 계약을 형식화하여 보안성에 대하여 증명하는 것이다. 퍼지 테스트는 입력으로 프로그램의 취약점을 노출시켜 탐지하는 방법이다. ML기반 방법은 최근 주목을 받고 있으며 스마트 계약의 바이트 코드 등에 존재하는 악의적인 행동의 패턴을 학습한 모델을 통하여 취약점을 공격하는 지에 대하여 판별하는 방법이다.

2.3 인공 신경망

인공 신경망은 생물의 뇌 속 뉴런의 작동 알고리즘을 본 따 만든 컴퓨팅 시스템이다. 인공 신경망은 다양한 계층으로 구성되며 각 계층은 입력층, 은닉층 그리고 출력층으로 구분된다. 각 계층은 생물학적 뉴런을 모방한 수많은 노드들로 구성된다. 각 계층의 노드들은 서로 연결되어 있으며, 활성화 함수를 내포한다. 이러한 활성화 함수를 거쳐 노드값이 계산되며 인공신경망은 피드포워드 형식으로 구성되어 있어 해당 노드값을 업데이트 해나가며 해당 값을 통하여 모델을 학습하게 된다.

3. 인공신경망 기반의 암호 분석 연구 동향

3.1 [1]

[1]에서는 스마트 계약의 세 가지 취약점인 has_short_address, has_flows 그리고 is_greedy에 중점을 두어 악의적인 계약을 탐지한다.

has_short_address는 계약이 예상보다 짧을 때 발생한다. 솔리디티는 누락된 바이트를 자동으로 0으로 채우게 되고, 악의적인 공격에 의하여 이러한 0이 앞으로 이동하게 되면 예금 금액 등에 포함되어 그 수치를 증가시키는 문제가 발생하게 된다.

is_greedy는 이더를 배포할 수 없는 스마트 계약을 greedy로 분류하게 된다. 이더리움에는 많은 3자 라이브러리 계약이 있으며 그 중 일부는 입출금 기능을 제공한다. 이때 이러한 라이브러리가 종료되면 라이브러리를 불러온 당사자는 인출하는 기능을 잃게 되며 계정의 잔액을 해제할 수 없게 된다.

has_flows는 오버플로 또는 언더플로를 탐지한다. 솔리디티는 8바이트에서 16바이트 정수로의 확장을 허용하지 않으므로 산술 연산으로 인한 오버플로 또는 언더플로가 발생하기 쉽다.

이러한 취약점에 대한 특징을 추출하는 새로운 방법인 슬라이싱 매트릭스를 제안하고, 각각의 취약점에 대한 취약점 탐지 모델을 구성한다. 이때 스마트 계약은 바이트 코드로 이루어져 있으므로 바이트 코드를 opcode로 디컴파일한 후, opcode의 기능을 추출하고 결합하여 슬라이스 행렬을 생성한다. 그 후, CNN(Convolution Neural network)을 사용하여 opcode의 특성을 추출하여 취약점을 탐지한다.

해당 모델은 f1-score를 통해 성능을 측정하였으며, 성능을 비교하기 위하여 총 세 가지의 모델을 사용하였다. 세 가지의 모델은 NNBOOF(Neural Network Based on Opcode Feature), CNNBOSM(Convolution Neural Network Based on Slice Matrix), RFBOOF(Random Forest Based on Opcode Feature)이다. has_short_address 측면에서 모델의 성능은 각각 0.88, 0.91, 0.95의 정확도 달성하였으며 is_greedy 측면에서는 0.81, 0.83, 0.91, 그리고 has_flows 측면에서는 0.71, 0.76, 0.85의 정확

도를 달성하였다. 결과적으로 랜덤포레스트를 활용한 취약점 탐지 기법이 가장 높은 성능을 보여주었다.

3.2 [5]

[5]에서는 다중 작업 학습 기반의 스마트 계약의 취약점을 탐지한다. 또한 방향성 있는 취약점 특징을 학습하기 위하여 보조 과제를 설정하였다. 해당 모델은 하단 공유 계층과 작업 특화 계층으로 구성된 하드 공유 디자인을 기반으로 한다. 우선 하단 공유 계층은 입력 계약의 의미 정보를 학습하며 작업 간의 해당 지식을 공유하는 방법을 결정하는 데에 사용한다. 해당 계층은 텍스트 표현은 단어 및 위치 임베딩에 의해 새로운 벡터로 변환한 후, 주의 메커니즘을 기반으로 하는 신경망을 사용하여 계약의 특징 벡터를 학습하고 추출한다. 다음으로 작업 특화 계층은 각 작업의 기능을 구현하기 위하여 사용된다. 공유 계층에서 특징을 학습하고 추출한 작업에 대한 분류 모델을 구성하기 위하여 CNN이 사용되었다. 해당 모델은 세 가지 유형의 취약점을 인식하며, 보조 취약점 탐지 작업을 추가한 후 취약점 유형을 더욱 잘 식별할 수 있음을 보여준다. 단일 작업 모델과는 다르게 다중 작업 모델은 동시에 여러 작업을 완료할 수 있으므로 시간, 연산 및 저장 면에서 단일 작업 모델보다 성능이 뛰어나며 비용이 적게 드는 것이 관찰되었다.

해당 모델은 f1-score를 통해 성능을 측정하였으며, 기존 논문에 비해 산술 취약성에 있어서 24.91%, 재진입 탐지에 있어서 20.73%가 향상되었다. 알 수 없는 주소 취약점에 대해서는 성능이 향상되었다는 것은 관찰되었으나, 기존 논문이 이러한 유형의 취약점 탐지를 지원하지 않아 비교 데이터를 제공하지 못하였다.

3.3 [6]

[6]에서는 스마트 계약 취약점 탐지를 위하여 GNN(Graph Neural Network)을 사용한다. 또한 데이터 및 제어 종속성에 따라 스마트 계약 기능의 구문 및 의미 구조를 표현하는 계약 그래프를 사용한다. 그래프의 노드는 함수 호출 또는 변수를 나타내고 간선은 임시 실행 추적을 나타낸다. 주요 노드를 강조 표시하기 위하여 그래프를 정규화하는 제거 단계를 설계하며, 정규화된 그래프에서 학습하기 위한 DR-GCN(Degree-free Graph Convolution Neural Network)과 프로그램 요소의 고유한 역할과 시간적 관계를 고려한 시간 메시지 전파 네트워크인 TMP(Temporal Message Propagation)를 제안한다.

해당 모델은 소스 코드에서 제어 흐름 및 데이터 흐름 의미론을 추출하고 폴맥 메커니즘을 모델링하는 그래프 생성 단계, 정규화 단계 부분 그래프, 취약점 모델링 및 탐지를 위한 메시지 전파 네트워크와 같은 3단계로 구성되어 있다.

3.4 [7]

[7]에서는 거래로부터 네 가지 행동 패턴을 식별하였으며 서로 다른 유형의 계약 간의 차이점을 구별하는 데에 사용하였다. 또한 실험 데이터 세트를 구성하기 위하여 스마트 계약을 슬라이싱하는 데이터 슬라이싱 알고리즘을 제안한다.

슬라이싱 알고리즘은 4단계로 구성된다. 우선 고정된 시간 T에 따른 타임 스텝을 설정하여 타임 스텝에서 생성된 데이터가 하나로 병합된다. 최소 데이터 길이 nMin과 최대 데이터 길이 nMax를 설정한다. 데이터의 수가 nMax를 만족하지 못하는 경우 부족한 데이터를 0으로 채우게 된다. 데이터의 수가 nMax를 초과하는 경우 데이터의 시작 부분에서 나머지 n이 nMin보다 작아질 때까지 일정 시간 간격마다 데이터를 슬라이싱한다. 이때 슬라이싱 된 데이터는 데이터의 양을 증가시키는 이점이 존재한다.

슬라이싱 알고리즘을 통해 시계열 조각을 얻은 후 해당 데이터에 LSTM (Long Short Term Memory)를 사용하여 모델을 학습한다.

해당 모델은 f1-score를 사용하여 성능을 측정하였으며 0.691 ~ 0.825의 성능을 달성하였다. 해당 모델의 데이터 세트가 적었으며 LSTM 네트워크의 훈련이 불충분하여 높지 못한 결과가 관측되었다. 또한 LSTM와 유사한 모델인 GRU(Gated Recurrent Units)에서는 LSTM에 비해 떨어지는 성능을 보여주었지만 약 25%의 적은 시간비용을 달성하였다.

4. 결론

스마트 계약의 취약점에는 산술 취약점, 재진입 취약점, 계약 간 취약점, 가스 관련 취약점 등 다양한 보안 문제가 존재하며 기존의 수동적인 취약점 탐지 방법들은 시간적 비용이 많이 소모되므로 인공신경망을 이용한 다양한 취약점 탐지 기법들이 제안되고 있다. 즉각적으로 이루어지는 스마트 계약 특성상 탐지 기법들 또한 실시간으로 이루어져야 하며

취약점 존재 여부만을 탐지하는 기존의 탐지 방법이 아닌 탐지 유형을 식별하는 등의 기술이 필요한 것으로 보인다. 또한 인공 신경망 특성상 목표로 하는 취약점만을 분류할 수 있기 때문에 해당 기능의 확장성을 넓힐 수 있는 방법 또한 연구되어야 할 것으로 보인다.

5. Acknowledgement

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00264, IoT 융합형 블록체인 플랫폼 보안 원천 기술 연구, 100%).

참고문헌

[1] Xing, Cipai, et al. "A new scheme of vulnerability analysis in smart contract with machine learning." *Wireless Networks* (2020): 1-10.

[2] Xu, Yingjie, et al. "A Novel Machine Learning-Based Analysis Model for Smart Contract Vulnerability." *Security and Communication Networks* 2021 (2021).

[3] Khan, Zulfiqar Ali, and Akbar Siami Namin. "A survey on vulnerabilities of Ethereum Smart Contracts." *arXiv preprint arXiv:2012.14481* (2020).

[4] Khan, Shafaq Naheed, et al. "Blockchain smart contracts: Applications, challenges, and future trends." *Peer-to-peer Networking and Applications* 14.5 (2021): 2901-2925.

[5] Huang, Jing, et al. "Smart Contract Vulnerability Detection Model Based on Multi-Task Learning." *Sensors* 22.5 (2022): 1829.

[6] Zhuang, Yuan, et al. "Smart Contract Vulnerability Detection using Graph Neural Network." *IJCAI*. 2020.

[7] Hu, Teng, et al. "Transaction-based classification and detection approach for Ethereum smart contract." *Information Processing & Management* 58.2 (2021): 102462.