[부산대학교 정보컴퓨터공학부]

2023 학년도 전기 졸업과제 착수보고서

과제명 - (블록체인 보안) Cross-contract 취약점 탐지 툴

지도교수 : 최윤호 교수님

팀명 : 컴쪽이 (팀 번호 : 51, 분과 : D)		
팀원 번호	**학번**	**이름**
1	201924437	김윤하
2	202055571	윤지원
3	202055616	최지원

목차

1. 과제 배경 및 목표

1-1. 배경 및 필요성

4차 산업혁명의 핵심 기술 중 하나인 블록체인은 분산 원장 기술로, 중앙화된 서버나 데이터베이스를 사용하지 않아 데이터의 변조나 위조를 방지할 수 있다. 블록체인은 보안성과 불변성 등의 특성으로 인해 금융, 물류 등 다양한 산업 분야에서 주목받고 있다.

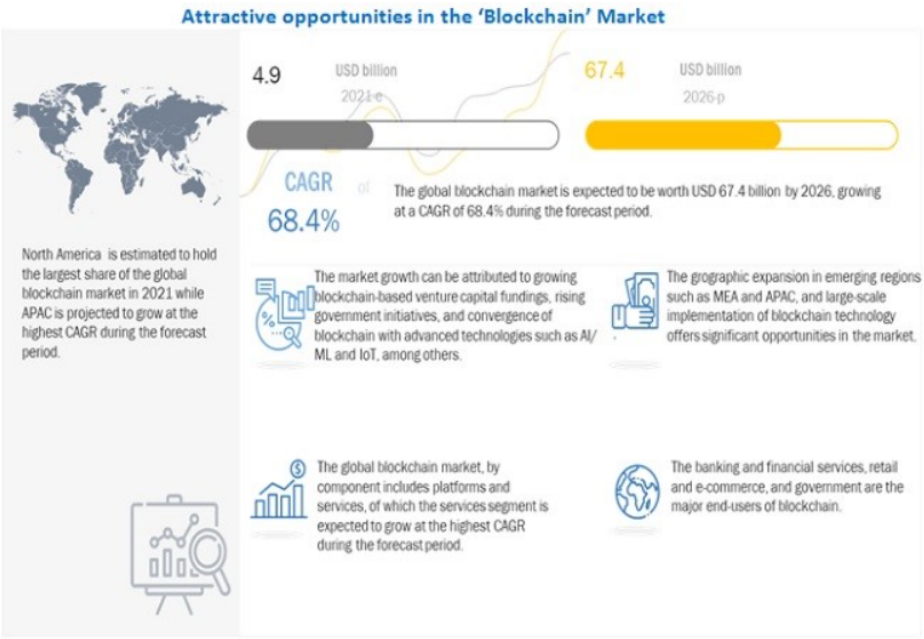


그림 1 블록체인 시장 규모 예측(자료: MarketsandMarkets)

시카고의 시장 조사 분석 기업 마켓앤마켓(MarketsandMarkets)이 '2026년까지 블록체인 시장 예측(Blockchain Market - Global Forecast to 2026)' 발표한 보고서에 따르면 블록체인 시장이 2026년까지 연간 68.4% 성장한다는 전망이다. 2021년 49억 달러로 예상되는 시장 규모는 2026년에 674억 달러로 크게 증가할 것으로 예상됐다.

블록체인 시장이 성장함에 따라 과학기술정보통신부는 국내 블록체인 기술 역량을 강화하기 위한 '데이터 경제를 위한 블록체인 기술 개발사업'의 수행기관 선정을 완료하여 기술개발에 착수한다고 밝혔다. 전략 분야 중 하나인 스마트 컨트랙트 보안 기술은 계약 조건이 충족되면 자동으로 거래를 수행하는 디지털 계약인 스마트 컨트랙트의 보안을 의미한다. 한 번 블록체인에 배포되면 수정할 수 없는 스마트 컨트랙트의 특성으로 인해 2016년 발생한 더 다오(The DAO) 사건과 같이 스마트 컨트랙트 코드상의 보안 취약점을 통한 악의적인 공격이 발생하고 있다.

▪ 사업기간/규모: '21년~'25년 / 총 1,133억원(국비 916억원, 민자 217억원)
※ '20.6월 예비타당성 조사 통과

▪ '21년 예산: 총 203.1억원(국고 182억원, 민자 21.1억원)

▪ 추진내용: 4대 전략분야 9대 기술 확보

그림 2 데이터 경제를 위한 블록체인 기술개발 사업 개요

한 번 배포되면 수정 불가한 특성을 고려했을 때, 스마트 컨트랙트 보안을 위해 배포되기 전 스마트 컨트랙트 코드상의 취약점을 탐지할 수 있는 도구가 필요하다. 이를 위해 본 과제에서는 인공지능을 기반으로 Static analysis 기법을 사용한 취약점 탐지 소프트웨어를 개발하고자 한다.

1-2. 국내외 기술 및 시장 현황

현재 제안된 인공지능 기반 스마트 컨트랙트 취약점 탐지 모델들은 다음과 같다.[1][2]

ContractWard	스마트 컨트랙트 소스코드를 바이트코드로 컴파일하여 preprocessing 과정을 거친 후, eXtreme Gradient Boosting, Adaptive Boosting, Random Forest, Support Vector Machine, k-Nearest Neighbor 다섯 가지 머신러닝 모델을 학습한 탐지 도구이다.
Combining Graph Neural Networks with Expert Knowledge	전문 지식과 인공지능 모델을 결합한 모델로 Combining graph feature and expert patterns를 통해 취약점을 탐지한다. 특정 취약점에 대한 전문 지식을 security pattern으로 만들고, 스마트 컨트랙트로부터 해당 패턴을 추출한다. 추출 후, 스마트 컨트랙트 소스코드를 범주화된 node와 edge로 표현한 contract graph를 생성하고 normalization한다. security pattern과 normalization이 수행

[1] 황도연; 김정구; 최윤호. "인공지능 기반 스마트 컨트랙트 취약점 탐지 연구 동향 분석", 한국통신학회 학술대회논문집, 2021, 66-67.

[2] 방지원; 최미정. "스마트 컨트랙트 취약점 탐지 도구 동향 분석, KNOM Review 25, no.1, 2022, 49-61.

	된 contract graph는 neural network를 통해 각각 벡터로 변환되며, 변환된 두 벡터는 결합되어 fully connected layer와 sigmoid layer를 통해 최종 취약점 탐지 결과로 계산된다. 전문 지식과 인공지능의 결합이 보다 정확한 스마트 컨트랙트 취약점 탐지를 가능하게 함을 보였다.
SMARTEMBED	기존 취약점 존재 코드와 코드 유사도를 측정하는 취약점 탐지 모델. 스마트 컨트랙트 소스코드를 Abstract syntax tree(AST)로 변환하고 이를 통해 소스코드를 contract-lever, function-level, statement-level의 세 가지 code snippet 형태로 재구성한 후, 취약점 탐지와 관련 없는 요소를 제거하는 normalization을 수행한다. normalization 처리된 데이터에 대해 code embedding learning을 수행한다. 이러한 embedding 과정을 통해 모든 code snippet은 동일한 크기의 벡터 표현으로 변환된다. 벡터 변환 과정이 수행된 이후 유사도 측정을 수행하여 유사도가 사전에 정해둔 임계치보다 높을 경우, 동일한 취약점이 존재하는 것으로 판단한다.
Eth2Vec	EVM의 바이트코드와 학습한 바이트코드 간의 유사성을 비교하여 취약점을 탐지하는 방식으로 구현된 SVM(Supprot Vector Machin)기반의 탐지 모델. 바이트코드를 벡터화해주는 EVM Extractor와 비지도학습 기반의 PV-DM 모델로 구성되어 있다.
SoliAudit	그레이박스기반의 퍼징 검사 기법과 머신러닝 모델을 결합한 탐지 도구. Dynamic fuzzer와 학습 모델로 구성된 Vulnerability analyzer로 각 취약점 유형에 대해 탐지를 동시에 수행하여 결과를 도출한다. Vulnerablitiy analyzer는 Logistic Regression, SVM, KNN, CNN 등 여러 종류로 구현하였고, 각 모델에 대해 취약점 탐지 유무 실험으로 성능 비교까지 검증하였다.
ESCORT	DNN(Deep Neural Network)기반으로 한 취약점 탐지 프레임워크. 취약점의 유형까지 분류할 수 있고, 새로운 취약점을 추가하기 위한 전이 학습을 지원한다. 학습을 위한 데이터셋 수집 및 레이블링을 자동으로 수행하는 ContractScraper과 취약점 유형 탐지를 위한 Multi Output Layer(MOL)-DNN 모델로 구성되어 있다.

1-3. 과제 목표

본 과제는 스마트 컨트랙트에서 Static analysis 기법을 사용한 취약점 탐지 소프트웨어 개발을 목표로 한다. CNN/GCN 모델을 기반으로 스마트 컨트랙트의 취약점인 Reentrancy, Access Control, Tx.origin, Time Manipulation을 탐지하고, 탐지될 경우 XAI 모델을 통해 발생 위치를 찾아 탐지된 취약점 종류 및 확률, 취약점 발생 위치를 보여주는 UI를 제공한다.

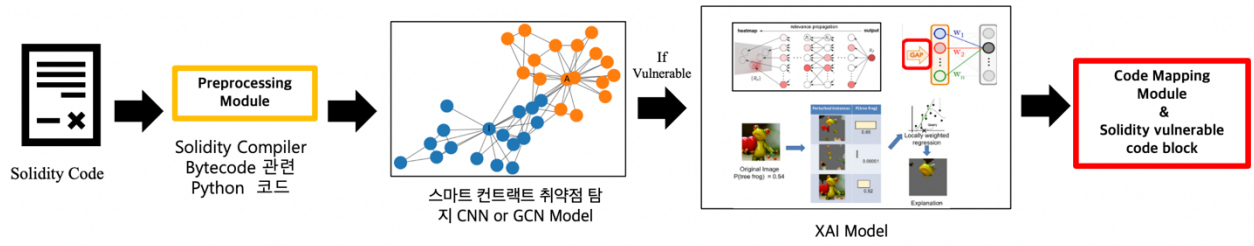
2. 세부 과제 내용

2-1. 과제 내용

우리가 개발하고자 하는 '(블록체인 보안) Cross Contract 취약점 탐지 소프트웨어'는 Smart Contract의 Reentrancy, Access Control, Tx.origin, Time Manipulation의 4가지 취약점의 패턴을 분석하여 어떤 Solidity Code가 취약점을 갖는지를 탐지하고, 그 Code Block을 사용자에게 알리는 모델이다.

이를 통해, 블록체인 네트워크상에서 중개자 없이 계약이 수행되는 Smart Contract의 취약점을 이용한 악용을 방지할 뿐만 아니라 프로그래머가 코드 작성 시 취약점에 유의해 코드를 작성할 수 있도록 돕고자 한다.

2-2. 전체 시나리오



[그림 3] 시스템 전체 구조

- User가 자신의 Solidity Code를 화면에 입력시, Python을 통해 Solidity Code를 Byte Code로 변환한다. 이 변환된 Byte Code를 CNN(이미지 처리) 또는 GCN(그래프 처리)으로 바꾼다. XAI Model을 통해 코드의 취약점을 파악해 Code Mapping Module을 통해 취약점을 갖는 Code Block의 위치를 Mapping해 User가 알아보기 쉽게 탐지된 취약 정보를 UI로 제공한다. 그 결과엔 탐지된 취약점 종류 및 확률, Solidity 파일의 CNN/GCN 결과, 취약점 발생 위치를 탐지한 Code Block이 포함된다.

2-3. 데이터 수집

1. 사용할 데이터 셋

학습 및 실험에 사용할 데이터 셋은 다음과 같다.

- Smartbugs
  - Smartbugs-wild
    - 이더리움 네트워크에서 추출한 47,398개의 스마트 계약이 포함되어 있다. 취약점에 대

해 레이블링 되어있지 않은 실제 계약 코드들을 기존 취약점 탐지 도구들을 활용해 4가지 취약점을 분석한다.

- Smartbugs-curated

  - 스마트 컨트랙트 취약점은 단순히 특정 코드를 보고 취약점을 판단하는 것이 아니라, 전체적인 코드의 문맥을 봐야 한다.[3]

  - 따라서 Smartbugs-curated 데이터 셋을 활용해, 기존 레이블링 된 취약 코드들을 성능 검증에 활용한다.

2. 사용할 취약점 탐지 툴[4]

다음의 취약점 탐지 도구들을 사용하며, 수동 검토와 함께 데이터 레이블링을 진행한다.

- **MythX**

  - 스마트 컨트랙트 보안 검증 플랫폼이다.

  - 실제 해킹 기법과 유사한 테스트를 수행하며 스마트 컨트랙트의 코드 상에 존재하는 보안 취약점을 식별한다.

- **Securify**

  - 이더리움 기반 스마트 컨트랙트의 취약점을 탐지하기 위한 자동 보안 분석 도구이다.

  - 정적 분석과 동적 분석 기법을 사용해 스마트 컨트랙트 코드에 존재하는 잠재적 취약점을 찾아낸다.

- **Oyente**

  - 이더리움 스마트 컨트랙트의 보안 검증 도구로, 정적 분석을 사용해 취약점을 찾는다.

- **SmartCheck**

  - Solidity 언어로 작성된 이더리움 스마트 컨트랙트의 보안 취약점을 검사한다.

  - 정적 분석 및 실제 시나리오를 시뮬레이션해 취약점을 식별한다.

2-4. 개발환경/기술스택

1. Python

Python	데이터 전처리, GCN or CNN, XAI model에 이용한다.
Scikit-learn	머신러닝 라이브러리로 오픈소스로 공개되어있다. 주요 기능으로는 기계학습 모델에 대해 설명하고 모델 선택을 위한 알고리즘을 제공한다

[3] 황선진. "이미지 Localization과 딥러닝 분류 기법을 활용한 스마트 컨트랙트 재진입 공격 취약점 위치 탐지 방법", 부산대학교 대학원 정보융합공학과 석사학위논문, 2020, 12.

[4] Github – ConsenSys, 이더리움 개발자 도구 리스트. https://github.com/ConsenSys/ethereum-developer-tools-list/blob/master/README_Korean.md

Numpy	수학적 연산과 다차원 배열 처리에 최적화된 라이브러리이며 데이터 분석, 머신러닝 분야에서 사용된다.
Scipy	수치해석기능을 제공하는 python 라이브러리이다.
Pandas	데이터 처리와 분석을 위한 라이브러리이며 대용량 데이터들을 처리할 때 유리하다

2. Solidity

Solidity	취약점 탐지를 위한 solidity file 분석에 이용한다. 이더리움에서 제공하는 스마트 컨트랙트 개발 언어이다. 정적 타입의 언어이며 EVM에서 구동되도록 설계되었다.
Truffle	이더리움 기반 DApp 개발을 도와주는 블록체인 프레임워크이다. 블록체인에서 스마트 컨트랙트를 개발, 배포, 테스트 환경을 제공해준다.

3. HTML, CSS, JavaScript

HTML, CSS, JavaScript	취약점 위치 시각화 UI 개발에 이용한다.
Chart.js	데이터 시각화를 위한 JavaScript 라이브러리이다.

4. MySQL, Git, VS Code

MySQL	오픈 소스 관계형 데이터베이스
git	개발 협력 및 프로젝트 관리
VS Code	개발 IDE

3. 과제 세부 요구사항 및 개발 내용

3-1. Smart Contract 취약점 종류[5]

1. Reetrancy (재진입)

스마트 컨트랙트는 보통 자신의 상태를 업데이트 한 후 다른 스마트 컨트랙트나 함수를 호출한다. 이 때 외부 계약 호출(External Contract Call)이 완료되기 전에 해당 계약 호출(자기 자신)을 다시 요청해 계약에 재진입이 가능한 경우이다. 이런 경우 공격자는 여러 번 자신의 계정에 대해 함수를 실행시키는 등의 공격을 할 수 있다. 'The Dao 해킹 사건'의 원인이 된 취약점이기도 한다. 이를 방지하기 위해서는, 보통 재진입 제어 코드를 작성해 한 번의 호출에서 함수가 완료되기 전에는 다시 호출하지 못하도록 해야 한다. 또한 call() 함수 사용을 자제하고, 외부 호출 함수를 내부 작업이 끝난 뒤에 실행해야 한다.

2. Access Control (AC, 접근 제어)

스마트 컨트랙트는 보통 여러 사용자가 참여하여 실행된다. 이 때, 각 사용자의 권한이 적절하게 관리되지 않으면 중요한 데이터나 자원에 대한 악용이 발생할 수 있다. 잘못된 접근 제어로 인해 접근 권한이 없는 공격자가 접근 권한을 부여받을 시, 접근제어에 의한 공격이 발생할 수 있다. 스마트 컨트랰트에서 접근 제어 취약점이 발생하는 경우는 다음 3가지가 있다.

1). Proxy library 및 proxy contract에서 delegatecall을 잘못 사용할 경우
2) 전역변수 tx.origin 변수로 인증 작업을 수행할 때
3) 긴 require() 함수를 처리할 때

1)로 인한 취약점이 발생한 실제 사례로는, 이더리움의 개빈우드(Gavin Wood)가 만든 이더리움 지갑, 패리티(Parity)가 해킹 당해 약 15만 Ether를 도난당한 사건이 있다. 해킹당한 방식은, delegatecall 함수를 통해 walletLibrary에 있는 initowner 함수를 public으로 호출이 가능하게 설정되어, 지갑의 권한이 넘어간 경우이다.

3. Tx.origin (주소)

공격자가 스마트 컨트랙트에 대한 transaction 을 전송 시, 해당 transaction 의 송신자는 Tx.origin 이 된다. 공격자는 이를 자신의 주소로 변경해 송신자가 되어, 공격을 시도할 수 있다. 위의 접근 제어 2)의 경우도 마찬가지로 tx.origin 과 관련된 취약점이라고 할 수 있다. 이를 방지하는 위한 방법으로는, msg.sender 를 사용하는 방안 등이 있다.

4. Time Manipulation (시간 조작)

스마트 컨트랙트는 보통 시간에 따라 특정 동작을 수행하도록 프로그래밍 되어 있다. 그 중, 특정 시간대에 선착순으로 거래가 가능하거나, 할인 이벤트 등의 시간적인 요소가 포함된 경우

[5] 방지원; 최미정. "스마트 컨트랙트 취약점 탐지 도구 동향 분석, KNOM Review 25, no.1, 2022, 49-61.

now 등을 통해 시간 계산이 가능하다. 공격자가 의도적으로 시간을 조작할 경우 예상하지 못한 결과를 초래할 수 있다.

이를 방지하기 위해서 스마트 컨트랙트에서는 시간 조작이 불가능하도록 적절한 방어 매커니즘을 구현해야 한다. 에를 들어, 외부에서 시간 정보를 가져오는 방식을 사용하거나, 블록체인에서 제공하는 블록 시간을 사용하는 등의 방법을 사용해 보안을 강화할 수 있다.

3-2. Proprocessing 동작 로직

이더리움의 스마트 컨트랙트는 주로 Solidity 파일로 작성되며 컴파일 된 바이트 코드가 EVM에서 실행된다. 정적 분석 도구를 이용해서 스마트 컨트랙트 취약점을 탐지하기 위해 컴파일 된 바이트 코드를 python 코드로 인코딩한다.

1. Solidity Code Compile

- 입력으로 받은 Solidity 코드 컴파일하여 byte code 생성한다.

2. Byte Code 전처리

- Python code로 인코딩하는 과정에서 byte code로 되어있는 데이터 전처리가 필요하다.
- Byte code에서 Opcode를 추출하고 의미와 기능을 분석한다.

3. Python Code 변환

- 학습에 이용하기 위해 전처리 된 byte code를 python code로 변환한다.
- CNN or GCN 모델의 학습 데이터로 쓰기 위해 변환된 python code를 이미지(CNN)로 바꾸거나 그래프 형식(GCN)으로 변환한다.

3-3. CNN/GCN 및 XAI 모델 설명

- CNN (Convolutional Neural Network)

이미지 처리에서 널리 사용되는 신경망 구조로, Convolution과 Pooling 레이어를 반복하여 사용하며, 이미지의 특징을 추출하고 분류하는 데 효과적이다. Convolution Layer에서는 이미지의 특징을 추출하고, Pooling Layer에서는 이미지의 크기를 줄이고 특정한 특징을 강조한다. 마지막으로 Fully Connected Layer에서는 추출한 특징이 무엇을 의미하는 데이터인지 분류하는 작업을 한다.

- GCN (Graph Convolutional Network)
그래프 데이터를 처리하기 위한 신경망 구조로, 그래프의 노드와 엣지를 특징 벡터로 표현하며, 이를 기반으로 그래프 데이터를 분류하거나 예측한다. 합성곱 연산을 그래프에 적용하여 추출한 특징으로 주변 노드와의 관계를 이용하여 노드의 특징을 추출한다

- XAI (Explainable Artificial Intelligence)
인공지능 모델의 동작원리를 해석하고, 그 결과를 쉽게 이해할 수 있도록 설명하는 기술이다. 딥러닝 모델 등 복잡한 인공지능 모델에서 사용되며, 모델의 입력과 출력 간의 연관성을 시각적으로 표현하거나, 모델이 각각의 결정을 내리는 과정을 설명하는 방식으로 동작한다. 이를 통해 모델의 예측 결과를 내리는 데 있어 어떤 특징이 중요하게 작용했는지를 파악할 수 있다.

3-4. Code Mapping Module 작동 원리

Code Mapping Module은 CNN, GCN, XAI model을 통해 얻은 취약점 결과를 이용하여 Solidity 파일에서 문제가 되는 code block 위치를 찾아 매핑해주는 역할을 한다.

[입력] 취약점으로 분류된 이미지나 그래프
[출력] Solidity code의 취약점 위치

1. 취약점으로 식별된 이미지 or 그래프를 입력으로 받는다.
2. 해당 취약점이 어느 위치에서 탐지되었는지 확인한다.
3. 취약점 위치와 일치하는 Opcode를 찾는다.
   - 기존의 Opcode를 이용한다.
4. Opcode와 대응하는 Solidity code를 출력한다.

3-5. 시각화 UI 요구사항

1. 탐지된 취약점 종류 및 확률
2. Solidity 파일 CNN 혹은 GCN
3. 취약점 발생 위치 및 탐지 매핑 위치

4. 개발 일정 및 역할 분담

4-1. 개발 일정

6월					7월					8월					9월			
1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주	5주	1주	2주	3주	4주
Smart contract 취약점 스터디																		
		서버 구축																
			Solidity Compiler -> byte code 모듈															
					취약점 data set 라벨링 작업													
						CNN/GCN model 구현												
								XAI Model feature 생성										
									Code mapping 모듈 개발									
										UI 디자인 설계								
											UI 기능 개발 구현							
												테스트 & 디버깅						
													오류 수정					
														최종 발표 & 보고서 준비				
																	포스터 제작	

4-2. 역할 분담

이름	역할 분담
김윤하	- Solidity Compiler 바이트 코드 관련 Python code 개발
윤지원	- GUI 시각화(HTML, javascript 활용)
최지원	- Code mapping module 개발 & Solidity vulnerable code block 생성
공통	- Cross-contract 에서 발생하는 Reentrancy, Access Control 등의 취약점 패턴 분석 - 기존 취약점 탐지 툴을 활용한 취약 Data set 레이블링 작업