안전한 스마트 컨트랙트를 위한 보안 약점 진단 도구에 관한 연구

김병현*, 김상수* 손윤식**, 이양선*
*서경대학교 컴퓨터공학과
**동국대학교 컴퓨터공학과
e-mail : byunghyun@skuniv.ac.kr

A Study on Security Weakness Dianostic Tool for Secure Smart Contract

Byunghyun Kim*, Sangsoo Kim* , Yunsik Son**, Yangsun Lee*
*Dept of Computer Engineering, SeoKyeong University
**Dept of Computer Science and Engineering, Dongguk University

요 약

블록체인을 활용한 기술들이 발전함에 따라 스마트 컨트랙트(smart contract)의 사용 또한 증가하고 있다. 블록체인 기반 스마트 컨트랙트가 디지털 경제에 혁신을 가져올 것으로 기대하지만 이 기술이 안정적으로 활용되기 위해서는 해결해야 할 보안 문제가 많다.

본 논문은 스마트 컨트랙트의 보안 약점을 컨트랙트 실행 전 사전에 분석하기 위해 체인 코드의 보안 약점 항목을 정의하였고, 이를 기반으로 보안 약점을 진단하는 이더리움 스마트 컨트랙트 진단 도구를 설계 및 구현하였다.

1. 서론

블록체인을 활용한 기술들이 발전함에 따라 스마트 컨트랙트의 사용 또한 증가하고 있다. 블록체인 기반 스마트 컨트랙트가 디지털 경제에 혁신을 가져올 것으로 기대하지만 이 기술이 안정적으로 활용되기 위해서는 해결해야 할 보안 문제가 많다. 그 중 하나로 최근 이더리움에서 DAO, Parity 해킹 사건이 발생함에 따라 스마트 컨트랙트에 대한 신뢰가 떨어지고 있다[1-2]. 따라서 이더리움 스마트 컨트랙트의 보안 약점 항목을 정의하고 검사하여 보안 위험을 예방해야 할 필요가 있다.

본 논문에서는 스마트 컨트랙트의 보안 약점을 컨트랙트 실행 전 사전에 분석하기 위해 체인 코드의 보안 약점 항목을 정의하였고, 이를 기반으로 보안 약점을 진단하는 이더리움 스마트 컨트랙트 진단 도구를 설계 및 구현하였다

2. 관련연구

2.1 SmartCheck

SmartCheck는 SmartDec에서 2018년에 개발한 스마트 컨트랙트 진단 도구다[3]. 표 1은 SmartCheck의 스마트 컨트랙트 보안 약점 항목을 나타낸다.

"이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.2019R1F1A1045343)"

표 1. SmartCheck의 스마트 컨트랙트 보안 약점 항목

2018.06.	2019.06.
1. No payable fallback function	1. No payable fallback function
2. Reentrancy	2. Compiler version not fixed
3. Unchecked math	3. Using tx.origin for authorization
4. Unchecked low-level call	4. Unchecked low-level call
5. Implicit visibility level	5. Implicit visibility level

2.2 SECURIFY

SECURIFY는 2017년 ETH Zurich(취리히 연방 공과대학교)의 블록체인 보안 프로젝트의 결과물로 이더리움 스마트 컨트랙트 진단 도구다[4]. SECURIFY는 정적, 동적 분석 등 자동화된 분석 기술을 통해 스마트 컨트랙트를 진단하고 그 결과를 사용자에게 전달하여 보안 위험을 예방한다. 표 2는 SECURIFY의 스마트 컨트랙트 보안 약점 항목을 나타낸다.

2018.06에 정의된 진단항목과 2019.06에 변경된 진단항목을 비교해보면 최근 보안 약점의 연구 방향과 중요도를 파악할 수 있다.

표 2. SECURIFY의 스마트 컨트랙트 보안 약점 항목

2018.06.	2019.06.
1. Transaction Reordering	1. Transaction Reordering
2. Recursive Calls	2. Recursive Calls
3. Insecure Coding Patterns	3. Insecure Coding Patterns
4. Unexpected Ether Flows	4. Unexpected Ether Flows
5. Use of Untrusted Inputs in Security Operations	5. Dependence on unsafe inputs

3. 보안 약점 진단기

본 논문은 스마트 컨트랙트의 보안 약점을 사전에 분석하기 위해 체인 코드의 보안 약점 항목을 정의하였고, 정의된 보안 약점 항목을 진단하는 보안 약점 진단기를 설계 및 구현하였다. 보안 약점 진단기는 스마트 컨트랙트의 추상 구문 트리(AST : Abstract Syntax Tree)를 정적 분석하여 진단한다. 그림 1은 보안 약점 진단기의 구조도를 나타낸 것이다.

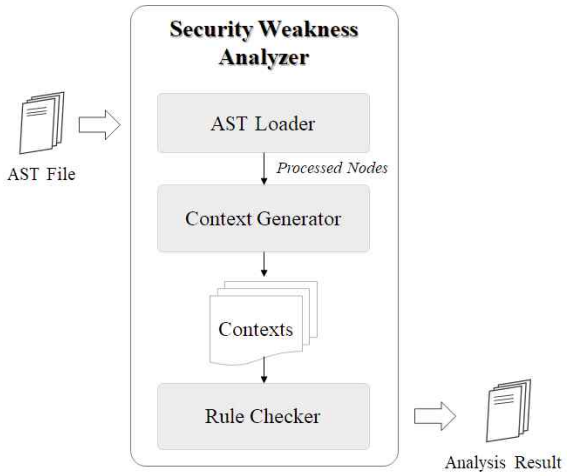

그림 1. 보안 약점 진단기 구조도

3.1 이더리움 스마트 컨트랙트 보안 약점 항목 정의

표 3. 제안하는 스마트 컨트랙트 보안 약점 항목

항목	내용
1. 데이터 검증	1) 정수형 오버플로우 & 언더플로우
2. 재진입성	1) Ether 전송 함수
	2) Fallback 함수
3. 예외 발생	1) 서비스 거부
4. 컨트랙트 특성	1) 다중 상속
	2) 접근지정자
5. 특수 변수	1) tx 변수

본 논문은 이더리움에서 스마트 컨트랙트를 실행할 때 발생할 수 있는 보안 약점 항목을 정의하였다. 보안 약점 진단기에서 아래와 같은 항목을 기반으로 스마트 컨트랙트를 진단하고 그 결과를 사용자에게 전달하여 보안 위험을 예방할 수 있다. 표 3은 본 논문에서 제안하는 이더리움 스마트 컨트랙트 보안 약점 항목을 나타낸다.

3.2 AST 로더

솔리디티 컴파일러에 의해 생성된 추상 구문 트리는 Json 형식의 파일로 생성된다. 보안 약점 진단기의 AST 로더는 솔리디티 컴파일러가 생성한 추상 구문 트리의 노드들의 필요한 정보들을 수집하여 사전에 정의된 노드 정보들로 가공한다. 가공된 노드들은 분석 비용을 줄이기 위한 콘텍스트 생성에 사용된다.

3.3 보안 약점 분석을 위한 콘텍스트 생성

추상 구문 트리를 이용한 정적 프로그램 분석은 불필요한 정보를 제거하고 분석하기 때문에 트리가 차지하는 메모리 공간이 절약된다. 하지만 각 항목을 진단할 때마다 보안 약점 진단을 위한 정보를 얻기 위해 트리를 순회하는 것은 비효율적이다.

콘텍스트 생성기(Context Generator)는 솔리디티 컴파일러가 생성하는 추상 구문 트리를 사전에 순회하여 진단에 필요한 데이터를 콘텍스트로 분류 및 저장한 후 사용한다. 이러한 방식은 보안 약점 진단 시간 비용을 줄일 수 있다. 표 4는 진단에 사용되는 콘텍스트 항목이다.

표 4. 보안 약점 진단에 사용되는 콘텍스트

항목	설명
1. Contract	컨트랙트 정의 노드 집합
2. Expression	표현식 노드 집합
3. Function Definition	함수 정의 노드 집합
4. Function Call	함수 호출 노드 집합
5. Variable	변수 선언 노드 집합

3.4 규칙 검사기

규칙 검사기(Rule Checker)는 콘텍스트 생성기에 의해 생성된 콘텍스트를 이용하여 정해진 규칙에 따라 정의된 보안 약점 항목을 검사한다.

3.4.1 재진입성 검사

Fallback 함수를 악용한 재진입성 공격 예방을 위해서는 Ehter 전송 함수호출 시 contract.call() 대신 transfer()를 사용해야 한다. 보안 약점 진단기는 Expression, FunctionCall 콘텍스트를 이용해 호출 하는 함수의 형태를 파악하여 진단한다. 사용자 정의 함수의 재진입성 검사의 경우에는 FunctionCall 콘텍스트에서 참조하는 함수의 id와 참조되는 함수의 id를 이용하여 진단할 수 있다. 그림 2는 함수호출 콘텍스트 정보를 이용한 재진입 진단 구조다.

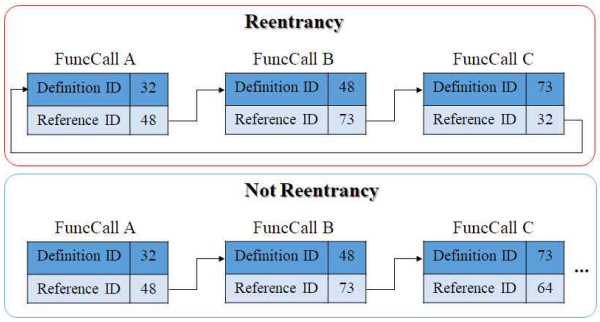

그림 2. 보안 약점 진단기의 재진입 진단

3.4.2 컨트랙트 특성 검사

솔리디티에서 컨트랙트는 우선 순위를 설정하는 역 C3 선형화(reverse C3 Linearization)를 사용하여 다중 상속 시 기능의 모호성을 처리한다. 컨트랙트의 우선 순위에 따라 기능이 달라질 수 있으므로 상속 순서가 매우 중요하다. 다중 상속의 모호성은 Contract, FunctionDefinition 콘텍스트에서 컨트랙트 상속 구조와 함수의 형태를 비교하여 부모 컨트랙트 중 같은 형태의 함수가 존재하는지 판단한다.

컨트랙트안에 존재하는 함수는 접근지정자를 작성하지 않을 경우 public으로 설정된다. Parity 해킹 사건과 관련된 내용이며 이러한 특성은 예상하지 못한 기능 실행을 불러올 수 있다. 보안 약점 진단기는 FunctionDefinition 콘텍스트를 이용하여 각 함수가 접근지정자를 사용했는지 검사한다.

4. 실험 결과

본 논문의 보안 약점 진단기는 스마트 컨트랙트 보안 약점 분류 및 테스트 케이스(Smart Contract Weakness Classification and Test Cases)에 정의된 보안 약점 중 SWC-107: 재진입성(Reentrancy), SWC-125: 잘못된 상속 순서(Incorrect Inheritance Order)를 검사한다.

표 5. SWC-107 예제 소스 코드 및 진단 결과

SimpleDao.sol

3: contract SimpleDAO {
4: mapping (address => uint) public credit;
5:
6: function withdraw(uint amount) public {
7: if (credit[msg.sender]>= amount) {
8: require(msg.sender.call.value(amount)());
9: credit[msg.sender]-=amount;
10: }
11: }
12: ... 중략 ...
13: }

Result
========== Reentrancy ========== [MAJOR] Reentrancy warning "Incorrect function usage in Ether transmission, Use transfer() instead", Line : 8

표 5는 SWC-107 예제 simple_dao.sol 소스 코드와 진단 결과이다. 소스 코드의 8번째 줄에서 withdraw 함수의 재귀 호출이 종료되지 않아 Ether를 계속 전송하는 현상이 발생할 가능성이 존재한다.

표 6. SWC-125 예제 소스코드 및 진단결과

MDTCrowdsale.sol

468: contract CappedCS is Crowdsale {
: ... 중략 ...
486: function validPurchase() internal constant returns (bool) {
: ... 중략 ...
489: }
490: }
492: contract WhitelistedCS is Crowdsale {
: ... 중략 ...
504: function validPurchase() internal constant returns (bool) {
: ... 중략 ...
506: }
507: }
509: contract MDTCrowdsale is CappedCS, WhitelistedCS {
: ... 중략 ...
519: }

Result
========== Multiple-Inheritance ========== [MAJOR] Multiple-Inheritance warning "Do not write multiple inheritance", Line : 509

표 6은 SWC-125 예제 MDTCrowdsale.sol 소스 코드와 진단 결과이다. MDTCrowdsale 컨트랙트가 상속받은 CappedCrowdsale, WhitelistedCrowdsale 컨트랙트에는 동일한 함수 validPurchase()가 존재하여 다중 상속 문제가 발생한다.

5. 결론 및 향후연구

블록체인을 활용한 기술들이 발전함에 따라 스마트 컨트랙트의 사용 또한 증가하고 있다. 블록체인 기반 스마트 컨트랙트가 디지털 경제에 혁신을 가져올 것으로 기대하지만 안정적으로 활용되기 위해서는 해결해야 할 보안 문제가 많다.

본 논문은 스마트 컨트랙트의 보안 약점을 실행 전 사전에 분석하기 위해 체인 코드의 보안 약점 항목을 정의하였고 보안약점 진단기를 구현하였다. 현재 구현된 보안 약점 진단기는 정의한 보안 약점 중 재진입과 다중 상속 항목만 진단이 가능하다. 앞으로 보안 약점 진단기가 나머지 보안 약점 항목들을 진단할 수 있도록 구현할 예정이다.

참고문헌

[1] Quinn Dupont, Experiments in Algorithmic Governance: A history and ethnography of “The DAO“, a failed Decentralized Autonomous Organization, 2017.

[2] SANTIAGO PALLADINO, “The Parity Wallet Hack Explained,” https://blog.zeppelin.solutions/on-the-parity-wallet-multisig-hack-405a8c12e8f7, 2017.

[3] “SmartCheck,” https://github.com/smartdec/smartcheck.

[4] “Securify,” https://github.com/eth-sri/securify.