

13th AUG 2021



WLOG

SMART CONTRACT AUDIT REPORT

version v1.0

ERC20 Security Audit and General Analysis

HAECHEI AUDIT

COPYRIGHT 2021. HAECHEI AUDIT. all rights reserved

Table of Contents

0 Issues (0 Critical, 0 Major, 0 Minor) Found

[Table of Contents](#)

[About HAECHI AUDIT](#)

[01. Introduction](#)

[02. Summary](#)

[Issues](#)

[03. Overview](#)

[Contracts Subject to Audit](#)

[Key Features](#)

[04. Issues Found](#)

[TIPS : 누락된 Event가 있습니다 \(Found - v.1.0\)](#)

[TIPS : 사용하지 않는 함수가 존재합니다 \(Found - v.1.0\)](#)

[05. Disclaimer](#)

[Appendix A. Test Results](#)

About HAECHI AUDIT

HAECHI AUDIT은 글로벌 블록체인 업계를 선도하는 HAECHI LABS의 대표 서비스 중 하나로, 스마트 컨트랙트 보안 감사 및 개발을 전문적으로 제공합니다.

다년간 블록체인 기술 연구 개발 경험을 보유하고 있는 전문가들로 구성되어 있으며, 그 전문성을 인정받아 블록체인 기술 기업으로는 유일하게 삼성전자 스타트업 육성 프로그램에 선정된 바 있습니다. 또한, 이더리움 재단과 이더리움 커뮤니티 펀드로부터 기술 장려금을 수여받기도 하였습니다.

HAECHI AUDIT의 보안감사 보고서는 전세계 암호화폐 거래소들의 신뢰를 받고 있습니다. 실제로 많은 클라이언트들이 HAECHI AUDIT 스마트 컨트랙트 보안감사를 거친 후에, Huobi, OKEX, Upbit, Bithumb 등에 성공적으로 상장하였습니다.

대표적인 클라이언트 및 파트너사로는 글로벌 블록체인 프로젝트와 포춘 글로벌 500대 기업들이 있으며, 카카오의 자회사인 Ground X, Carry 프로토콜, Metadium, LG, 한화, 신한은행 등이 있습니다. 지금까지 약 60여곳 이상의 클라이언트를 대상으로 가장 신뢰할 수 있는 스마트 컨트랙트 보안감사 및 개발 서비스를 제공하였습니다.

문의 : audit@haechi.io

웹사이트 : audit.haechi.io

01. Introduction

본 보고서는 WLOG 팀이 제작한 WLOGToken 스마트 컨트랙트의 보안을 감사하기 위해 작성되었습니다. HAECHI AUDIT 는 WLOG 팀이 제작한 스마트 컨트랙트의 구현 및 설계가 공개된 자료에 명시한 것처럼 잘 구현이 되어있고, 보안상 안전한지에 중점을 맞춰 감사를 진행했습니다.

발견된 이슈는 중요도 차이에 따라 **CRITICAL**, **MAJOR**, **MINOR**, **TIPS**로 나누어집니다.

CRITICAL

Critical 이슈는 광범위한 사용자가 피해를 볼 수 있는 치명적인 보안 결점으로 반드시 해결해야 하는 사항입니다.

MAJOR

Major 이슈는 보안상에 문제가 있거나 의도와 다른 구현으로 수정이 필요한 사항입니다.

MINOR

Minor 이슈는 잠재적으로 문제를 발생시킬 수 있으므로 수정이 요구되는 사항입니다.

TIPS

Tips 이슈는 수정했을 때 코드의 사용성이나 효율성이 더 좋아질 수 있는 사항입니다.

HAECHI AUDIT는 WLOG 팀이 발견된 모든 이슈에 대하여 개선하는 것을 권장합니다.

이어지는 이슈 설명에서는 코드를 세부적으로 지칭하기 위해서 {파일 이름}#{줄 번호}, {컨트랙트 이름}#{함수/변수 이름} 포맷을 사용합니다. 예를 들면, `Sample.sol:20`은 Sample.sol 파일의 20번째 줄을 지칭하며, `Sample#fallback()`는 Sample 컨트랙트의 fallback() 함수를 가리킵니다

보고서 작성을 위해 진행된 모든 테스트 결과는 Appendix에서 확인하실 수 있습니다.

02. Summary

Audit에 사용된 코드는 Etherscan

(<https://etherscan.io/address/0x79CEe0e52295d2d1Ea7eBc51f9C16aa31F9dD74a#code>)에서 찾아볼 수 있습니다.

Issues

HAECHI AUDIT에서는 Critical 이슈 0개, Major 이슈 0개, Minor 이슈 0개를 발견하였으며 수정했을 때 코드의 사용성이나 효율성이 더 좋아질 수 있는 사항들을 2개의 Tips 카테고리로 나누어 서술하였습니다.

Severity	Issue	WLOGTokenus
TIPS	누락된 Event가 있습니다	(Found - v1.0)
TIPS	사용되지 않는 함수가 있습니다	(Found - v1.0)

03. Overview

Contracts Subject to Audit

- Context
- SafeMath
- ERC20
- ERC20Burnable
- WLOGToken

Key Features

WLOG 팀은 아래의 기능을 수행하는 ERC20 Smart contract를 구현하였습니다.

- 토큰 소각(Burn)

04. Issues Found

TIPS : 누락된 Event가 있습니다 (Found - v.1.0)

TIPS

다음은 Event가 누락된 함수들의 목록입니다.

Function	Expected Event	Emitted Event	Omitted Event
burn	Transfer, Burn	Transfer	Burn
burnFrom	Transfer, Approval, Burn	Transfer, Approval	Burn

Event가 없는 경우, 블록체인에 정확한 값의 기록 여부를 실시간으로 파악하는데 어려움이 있습니다. 이 경우, application에서 해당 값의 변경 여부 및 해당 함수 호출 여부를 파악하기 어렵습니다.

따라서, 해당 함수에서 발생하는 변화에 해당하는 Event를 추가하시는 것을 추천드립니다.

TIPS : 사용하지 않는 함수가 존재합니다 (Found - v.1.0)

TIPS

```
26 - function _msgData() internal view virtual returns (bytes memory) {
27     this; // silence state mutability warning without generating bytecode - see https://github.com/ethereum/solidity/issues/2691
28     return msg.data;
29 }
30 }
```

[<https://etherscan.io/address/0x79CEe0e52295d2d1Ea7eBc51f9C16aa31F9dD74a#code#L26>]

`Context#_msgData()` 함수가 사용되지 않습니다.

05. Disclaimer

해당 리포트는 투자에 대한 조언, 비즈니스 모델의 적합성, 버그 없이 안전한 코드를 보증하지 않습니다. 해당 리포트는 알려진 기술 문제들에 대한 논의의 목적으로만 사용됩니다. 리포트에 기술된 문제 외에도 이더리움, 솔리디티 상의 결함 등 발견되지 않은 문제들이 있을 수 있습니다. 안전한 스마트 컨트랙트를 작성하기 위해서는 발견된 문제들에 대한 수정과 충분한 테스트가 필요합니다.

Appendix A. Test Results

아래 결과는, 보안 감사 대상인 스마트 컨트랙트의 주요 로직을 커버하는 unit test 결과입니다. 붉은색으로 표시된 부분은 이슈가 존재하여 테스트에 통과하지 못한 테스트 케이스입니다.

SafeMath

add

- ✓ adds correctly
- ✓ reverts on addition overflow

sub

- ✓ subtracts correctly
- ✓ reverts if subtraction result would be negative

mul

- ✓ multiplies correctly
- ✓ multiplies by zero correctly
- ✓ reverts on multiplication overflow

div

- ✓ divides correctly
- ✓ divides zero correctly
- ✓ returns complete number result on non-even division
- ✓ reverts on division by zero

mod

- ✓ reverts with a 0 divisor

modulos correctly

- ✓ when the dividend is smaller than the divisor
- ✓ when the dividend is equal to the divisor
- ✓ when the dividend is larger than the divisor
- ✓ when the dividend is a multiple of the divisor

WLOG Token

#constructor()

- ✓ name set properly
- ✓ symbol set properly
- ✓ decimals set properly
- ✓ initialSupply set properly
- ✓ totalSupply set properly

ERC20 Spec

#approve()

- ✓ should fail if spender is AddressZero
- ✓ should fail if owner is AddressZero

valid case

- ✓ allowance should set appropriately
- ✓ should emit Approval event

#transfer()

- ✓ should fail if recipient is AddressZero
- ✓ should fail if sender's amount is more than balance

valid case

- ✓ sender's balance should decrease
- ✓ recipient's balance should increase
- ✓ should emit Transfer event

#transferFrom()

- ✓ should fail if sender is AddressZero
- ✓ should fail if recipient is AddressZero
- ✓ should fail if sender's amount is more than balance
- ✓ should fail if sender's amount is more than allowance
- ✓ should fail if try to transfer sender's token without approve process

valid case

- ✓ sender's balance should decrease

✓ recipient's balance should increase

✓ allowance should decrease

✓ should emit Transfer event

✓ should emit Approval event

#increaseAllowance()

✓ should fail if overflows

valid case

✓ should increase allowance properly

✓ should emit Approval event

#decreaseAllowance()

✓ should fail if overflows

valid case

✓ should decrease allowance properly

✓ should emit Approval event

ERC20Burnable spec

#burn()

✓ should fail if try to burn more than burner's balance

valid case

✓ burner's balance should decrease

✓ totalSupply should decrease

✓ should emit Transfer event

1) should emit Burn event

#burnFrom()

✓ should fail if burnFrom addressZero

✓ should fail if value is greater than allowance

✓ should fail if value is greater than balance

valid case

✓ allowance should decrease

✓ should emit Approval event

✓ burner's balance should decrease

✓ totalSupply should decrease

✓ should emit Transfer event

2) should emit Burn event

Unused

✓ #_msgData()

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/					
WLOGToken.sol	100	100	100	100	

[丑 1] Test Case Coverage