KAKI TOKEN SECURITY ASSESSMENT REPORT

13.06 - 17.06.2019

Before start

- This document is based on vulnerability testing conducted by a blockchain security/development company, Differz, and focuses on the discovery of security vulnerabilities. In addition, we will discuss code quality and code license violations.
- This document does not guarantee or describe the usefulness of the code, the stability of the code, the suitability of the business model, the legal regulation of the business, the suitability of the contract, the bug, and the state of the vulnerability. Audit documents are used only for discussion purposes.
- We disclose that we have done our best to analyze smart contracts and do not disclose company information that we acquired during the course of our business or collect it through separate media.

Summary

From June 13, 2019 to June 17, 2019, DIFFERZ conducted a vulnerability analysis on "KAKI TOKEN."

During the audit period, the following actions have been carried out:

- Detect and analyze vulnerabilities with DIFFERZ's own vulnerability tester
- Results and analysis of open vulnerability analyzer Mythrill
- Detect and analyze vulnerabilities using the Smart Contract Static Vulnerability Analyzer released by Luniverse and Sooho.
- Create code modification recommendations based on direct code best practices and secure coding guides.

Blockchain security experts analyzed the vulnerability of KAKI Token. The participating security experts have excellent hacking skills and experience, winning prizes in domestic and overseas hacking competitions and conducting a number of CFPs about blockchain.

We scanned the generally known vulnerability signature from the KAKI TOKEN projectile and conducted a more complex security vulnerability inspection process using mythril, a useful security tool used primarily by the Ethereum community.

A total of 0 security vulnerabilities were encountered. And It is also a code developed using open-zeppelin, which was developed according to development best practices. We've also confirmed that all issues are resolved. We recommend that you conduct a continuous code audit to stabilize the service and analyze potential vulnerabilities

Analysis target

- Project Name : KAKI Token
- File Name : kaki.sol
- MD5 Hash : 23bd1ca96590d349f822e53a45d6d98c
- # of line : 374
- # of character : 12,873

Token Information

Token Name	KAKI TOKEN
Token Symbol	KKT
Decimial	18
Mintable	Yes
Burnable	Yes

Analysis Result

We analyzed it through Mythril and our own vulnerability analyzer, Sooho's vulnerability analyzer, **but we couldn't find any vulnerabilities.** This project is the **same as Open-zepplane's Pauseable token project, is an authorized direct because there are no changes, and all mathematical operations are free from problems using Open-zepplane's Safe Math library.**

However, we confirmed that there is no possibility of an integer overflow in the actual operation, although mythril will treat all the parts containing the "=" (equal sign) of the Safe Math library as a warning.

However, there were a few bugs that could occur in the solar compiler due to the low solitude version, but **no real impact has been identified on this code**. in that part

SignedArrayStorageCopy (low/medium-severity),

DynamicConstructorArgumentsClippedABIV2 (every low-severity),

UninitializedFunctionPointerInConstructor (any low-severity),

There are three, but there are no storage-related leakage issues, and the function pointer clearly dictates the designated areas, so there is no problem.