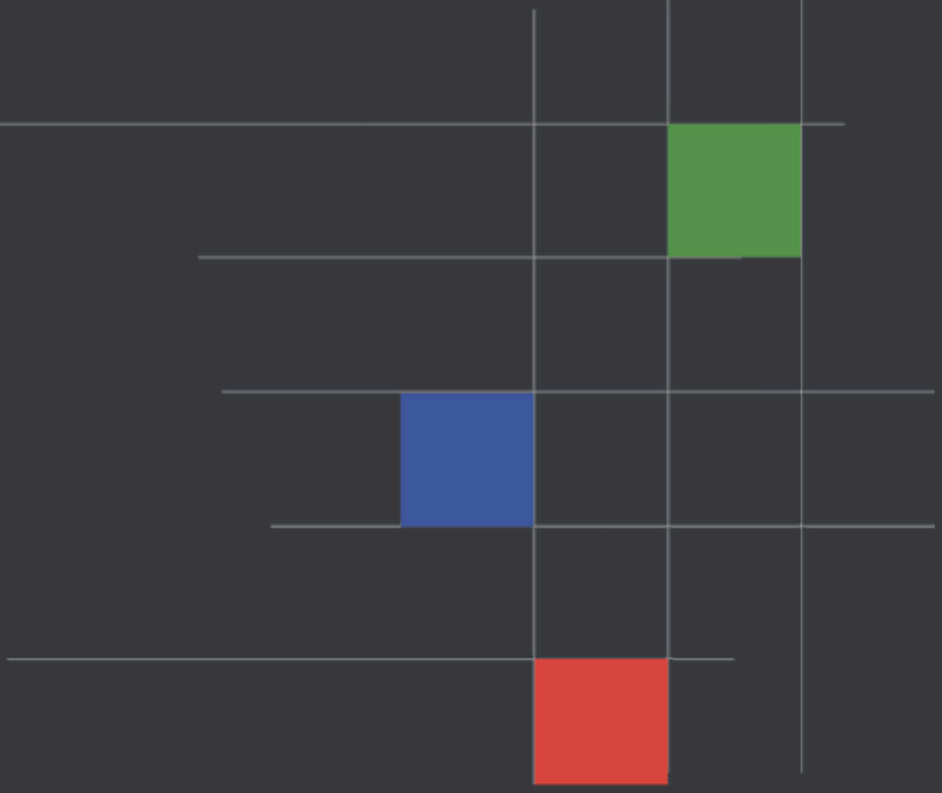
Tokocrypto Token

Security Assessment

March 31th, 2021

For:

Tokocrypto



Disclaimer

CertiK reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK’s position is that each company and individual are responsible for their own due diligence and continuous security. CertiK’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has indeed completed a round of auditing with the intention to increase the quality of the company/product’s IT infrastructure and or source code.

Overview

Project Summary

Project Name	Tokocrypto
Description	DeFi
Platform	Ethereum; Solidity
Codebase	GitHub Repository
Commit	e42d54bb49bff31f9f53b9dc4e2f2f82603354fc f2d57cc48c38d1734928cb5a347cd2b7eaa85b0a

Audit Summary

Delivery Date	Mar. 31th, 2021
Method of Audit	Static Analysis, Manual Review
Consultants Engaged	2
Timeline	Mar. 23th, 2021 - Mar. 27th, 2021, Mar. 31th, 2021

Vulnerability Summary

Total Issues	2
Total Critical	0
Total Major	1
Total Medium	0
Total Minor	1
Total Informational	0
Total Discussion	0
Notes	[**TKOToken.sol**]: Centralization issue is found in the contract TKOToken.sol, owner can mint and burn TKO token with desired amount, but not exceed the total supply.

Executive Summary

This report has been prepared for **Tokocrypto Token** smart contract to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Dynamic Analysis, Static Analysis, and Manual Review techniques.

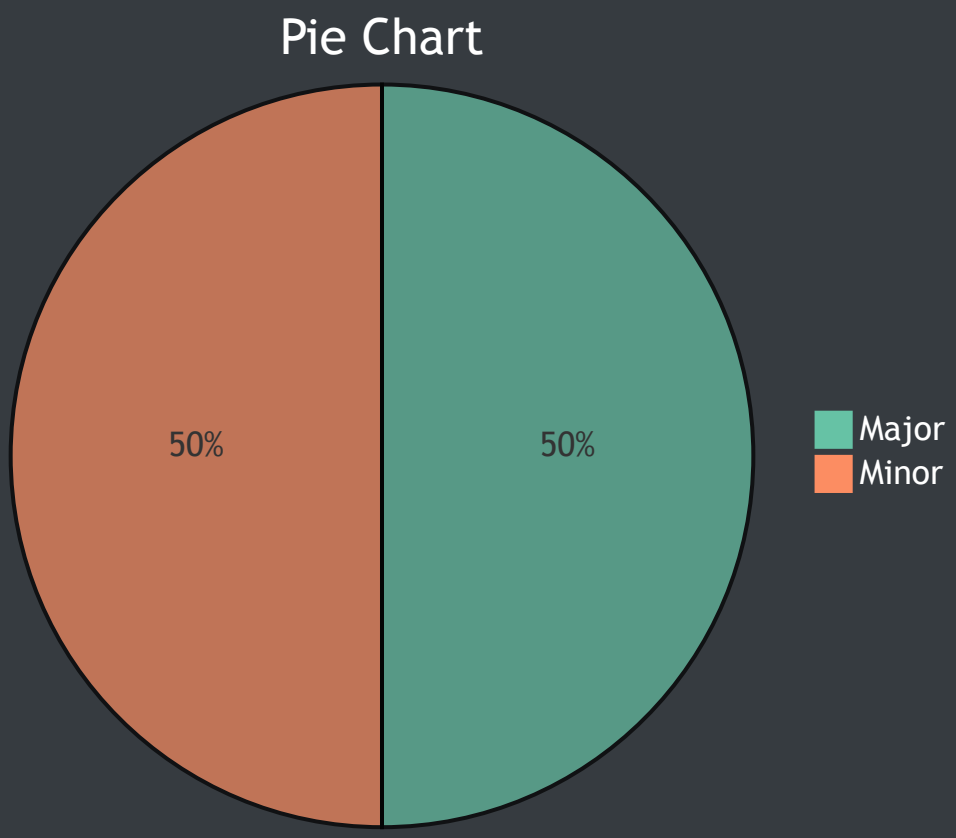
The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

File in Scope

ID	Contract	SHA-256 Checksum
TKO	TKOToken.sol	e44e6f47042b3f4bb4f8b42eba9b0fd923b96424bda08e753b0ee58963dded79
CTX	Context.sol	fac108458f6b48f68d6885f37010198a2eee1b6e37cb22c09d8dad7a31128e44
MGE	Manageable.sol	7e3377fe59d7f7a8aaf87279f8dc24d7e14c8dbfeb5a2cd03ceb2b80001fe713
OWN	Ownable.sol	5adde2ebe67492d7a8069425fac664d9566d2e8bc6bcf46a1d73bfc4df2d4f2e
SMT	SafeMath.sol	bdd6ad98b0f60db851928315e59092760a19f1241214d2e14b30a3506b81b2eb
BEP	BEP20.sol	523473d63886da2c45b3e2bb7d54f71445770bc062d71ddcad126035fbadba1a
SBEP	SafeBEP20.sol	27350d987cf13538639f76a24e6a3ef71d11394ff306281f9f60c43b8a8aec33
ADDR	Address.sol	4cc6a233f38c58f41f10cac8dacc91230027ac706cc59e2a23fc340399de7789

Findings



ID	Title	Type	Severity	Resolved
TKO-01	Incorrect burn flow	Logical Issue	Major	✓
TKO-02	Centralized Risk	Logical Issue	Minor	ⓘ

TKO-01: Incorrect burn flow

Type	Severity	Location
Optimization	Major	TKOToken.sol

Description:

Whenever new tokens are minted, new delegates are moved from the zero address to the recipient of the minting process. Hence, whenever tokens are burned, new delegates are once again moved from the zero address to the recipient whereas delegates should be moved on the opposite way.

Otherwise the delegate amount will be incorrect under governance mechanism.

Recommendation:

We advise that burn method should act as following:

1. burn the amount from `_from` address
2. move the delegate amount from address(0) to _delegates's _from address

```
function burn(address _from ,uint256 _amount) public onlyOwner {
    _burn(_from, _amount);
    _moveDelegates(address(0), _delegates[_from], _amount);
}

```

Alleviation:

[Tokocrypto Team]: The issue is addressed and reflected in the commit f2d57cc48c38d1734928cb5a347cd2b7eaa85b0a

TKO-02: Centralized Risk

Type	Severity	Location
Logical Issue	🔵 Minor	TKOToken.sol

Description:

`owner` is an important role in the contract. The owner address can operate on following sensitive functions:

- `mint()`
- `burn()`

Recommendation:

We advise the client to carefully manage the project's private key and avoid any potential risks of being hacked. We also advise the client to adopt Multisig, Timelock, and/or DAO in the project to manage sensitive role, `manager` in this case.

Appendix

Finding Categories

Gas Optimization

Gas Optimization findings refer to exhibits that do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Mathematical Operations

Mathematical Operation exhibits entail findings that relate to mishandling of math formulas, such as overflows, incorrect operations etc.

Logical Issue

Logical Issue findings are exhibits that detail a fault in the logic of the linked code, such as an incorrect notion on how `block.timestamp` works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Data Flow

Data Flow findings describe faults in the way data is handled at rest and in memory, such as the result of a `struct` assignment operation affecting an in-memory `struct` rather than an instorage one.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of `private` or `delete` .

Coding Style

Coding Style findings usually do not affect the generated byte-code and comment on how to make the codebase more legible and as a result easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a `constructor` assignment imposing different `require` statements on the input variables than a setter function.

Magic Numbers

Magic Number findings refer to numeric literals that are expressed in the codebase in their raw format and should otherwise be specified as `constant` contract variables aiding in their legibility and maintainability.

Compiler Error

Compiler Error findings refer to an error in the structure of the code that renders it impossible to compile using the specified version of the project.

Dead Code

Code that otherwise does not affect the functionality of the codebase and can be safely omitted.

Icons explanation

 : Issue resolved

 : Issue not resolved / Acknowledged. The team will be fixing the issues in the own timeframe.

 : Issue partially resolved. Not all instances of an issue was resolved.