



Mr.Fox Token
Smart Contract Audit Report

Prepared By: Sutee Sudprasert

ThaiChain, Thailand
August 31th, 2021

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to: (i) cybersecurity vulnerabilities and issues in the smart contract source code analysed, the details of which are set out in this report, (Source Code); and (ii) the Source Code compiling, deploying and performing the intended functions. In order to get a full view of our findings and the scope of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and ThaiChain and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (ThaiChain) owe no duty of care towards you or any other person, nor does Losiro make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and ThaiChain hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, ThaiChain hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against ThaiChain, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

Document Properties

| | |
|-------------------------|---|
| Client | Mister Fox Co., Ltd. |
| Title | Smart Contract Audit Report |
| Contract Address | https://bscscan.com/address/0xc2d59082aca32ff289da772756a01120fa9f6346 |
| Author | Sutee Sudprasert |
| Auditors | Sutee Sudprasert |
| Reviewed By | Thanarat Kuawattanaphan |
| Approved By | Dom Charoenyos |
| Classification | Confidential |

Introduction

Thai Chain was contracted by Mister Fox Co., Ltd. to conduct an audit of smart contracts. The report presents the findings of the security assessment of the smart contracts and its code review conducted at August 31th, 2021

Scope

The scope of the project is smart contracts in the address:

<https://bscscan.com/address/0xc2d59082aca32ff289da772756a01120fa9f6346>

Executive Summary

FOX token is a ERC20 standard token with 2,000M limited supply. Only the contract's minters will be able to mint the token and the contract's burners will be able to burn the token from other accounts. The minters and burners can only be assigned by the contract's admin. Additionally, the FOX token has `claimToken` function to rescue who accidentally transfers a token to the contract.

Our team performed static analysis, code functionality and manual audit. We found 3 issues during the audit.

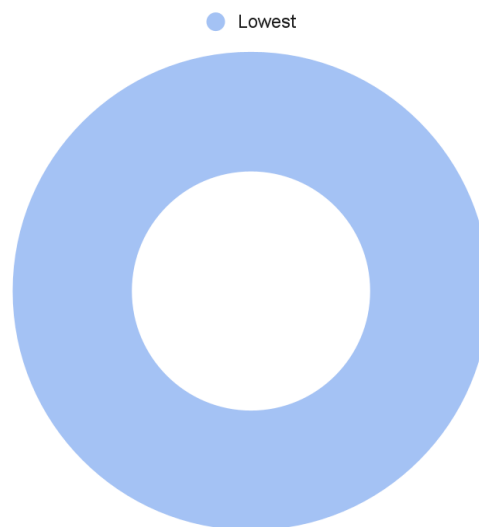
Severity Definitions

| Severity Level | Description |
|---------------------------------------|---|
| Critical | Critical vulnerabilities are usually straightforward to exploit and can lead to asset loss or data manipulations. |
| High | High-level vulnerabilities have a significant impact on smart contract execution, e.g., public access to crucial functions. |
| Medium | Medium-level vulnerabilities are important to fix; however, they can't lead to asset loss or data manipulations. |
| Low | Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution. |
| Lowest / Coding Style / Best Practice | Lowest-level vulnerabilities, code style violations, and info statements can't affect smart contract execution and can be ignored. |

Findings

3 Total Issues

Critical 0 (0%)
High 0 (0%)
Medium 0 (0%)
Low 0 (0%)
Lowest 3 (100%)



| ID | Title | Category | Severity | Status |
|--------|---|-------------------|----------|--------------|
| FOX-01 | compiler version not locked | Language Specific | Lowest | Acknowledged |
| FOX-02 | Public function that could be declared external | Gas Optimization | Lowest | Acknowledged |
| FOX-03 | Function name not follow naming conventions | Coding Style | Lowest | Acknowledged |

Audit Overview

Critical

No critical issues were found.

High

No critical issues were found.

Medium

No critical issues were found.

Low

No low issues were found.

Lowest / Coding Style / Best Practice

1. **FOX-01:** An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can also lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

Recommendation

Avoid a floating prama version (i.e. `pragma solidity ^0.8.0;`) instead specify a pragma version without using the caret symbol, i.e., `pragma solidity 0.8.0;`.

We recommend locking the compiler at the lowest possible version that supports all the capabilities wished by the codebase. This will ensure that the project utilizes a compiler version that has been in use for the longest time and as such is less likely to contain yet-undiscovered bugs.

2. **FOX-02:** `public` functions that are never called by the contract should be declared `external` to save gas.

- `mint(address,uint256)`
- `burnfrom(address,uint256)`
- `claimToken(IERC20)`
- `cap()`

Recommendation

Use the `external` attribute for functions never called from the contract.

3. **FOX-03:** According to [Solidity Style Guide](#), function names should be used `mixedCase`.

Recommendation

`burnfrom` function should be renamed to `burnFrom`.