AFIN Token

Smart Contract Audit Report

Prepared By: Sutee Sudprasert

ThaiChain, Thailand

August 11th, 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 Iosiro 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	Asian Fintech
Title	Smart Contract Audit Report
Contract Address	https://etherscan.io/address/0xee9e5eff401ee921b138490d00ca8d1f13f67a72
Author	Sutee Sudprasert
Auditors	Sutee Sudprasert
Reviewed By	Thanarat Kuawattanaphan
Approved By	Dom Charoenyos
Classification	Confidential

Introduction

Thai Chain was contracted by Asian Fintech 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 11th, 2021

Scope

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

https://etherscan.io/address/0xee9e5eff401ee921b138490d00ca8d1f13f67a72

Executive Summary

AFIN token is a ERC20 standard token with unlimited supply. Only the contract's owner will be able to mint. The ownership can be transferred to another address and can be renounced. The token can be burnt by the holder only. The token also has the ability to finish minting means that if `mintingFinished` was set, the token will not be able to mint anymore.

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

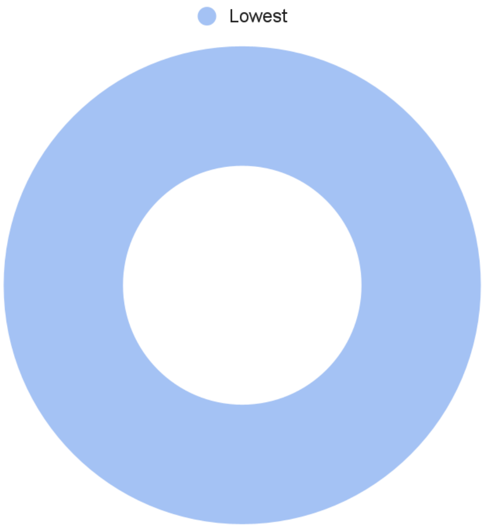
Severity Definitions

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
AFIN-01	Old compiler version declaration and version not locked	Language Specific	Lowest	Acknowledged
AFIN-02	Public function that could be declared external	Optimization	Lowest	Acknowledged
AFIN-03	State variables that could be declared constant	Optimization	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. **AFIN-01**: `solc` frequently releases new compiler versions. Using an old version prevents access to new Solidity security checks. We also recommend avoiding complex `pragma` statements.

   Recommendation
   Avoid a floating prama version (i.e. `pragma solidity ^0.4.23;`) instead specify pragma version without using the caret symbol, i.e., `pragma solidity 0.6.12;`.

   Deploy with any of the following Solidity versions:
   - 0.5.16 - 0.5.17
   - 0.6.11 - 0.6.12
   - 0.7.5 - 0.7.6. Use a simple pragma version that allows any of these versions.

   We recommend using the latest version of Solidity for testing.

2. **AFIN-02**: `public` functions that are never called by the contract should be declared `external` to save gas.
    - `totalSupply()`
    - `balanceOf(address)`
    - `transfer(address,uint256)`
    - `allowance(address,address)`
    - `transferFrom(address,address,uint256)`
    - `approve(address,uint256)`
    - `increaseApproval(address,uint256)`
    - `decreaseApproval(address,uint256)`
    - `renounceOwnership()`
    - `transferOwnership(address)`
    - `mint(address,uint256)`
    - `finishMinting()`
    - `burn(uint256)`

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

3. **AFIN-03**: Constant state variables should be declared constant to save gas.
    - `Afin.decimals`
    - `Afin.name`
    - `Afin.symbol`

    Recommendation
    Add the `constant` attributes to state variables that never change.