verichains

SECURITY AUDIT OF

FAIR TOKEN

Public Report

Mar 26, 2025

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
BSC	Binance Smart Chain or BSC is an innovative solution for introducing interoperability and programmability on Binance Chain.
BNB	A cryptocurrency whose blockchain is generated by the Binance Smart Chain platform. BNB is used for payment of transactions and computing services in the Binance Smart Chain network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report was prepared by Verichains Lab on Mar 26, 2025. We would like to thank the Fair and Free for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the Fair Token. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified no vulnerable issue in the smart contract code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY

1.1. Audit Scope

This audit focused on identifying security flaws in code and the design of the Fair Token. It was conducted on commit `142649ed1fcbf16789a2859247112df812571e0b` from git repository: https://github.com/fairfairfair3x/Fair3_BEP20/

SHA256 Sum	File
`447334e469eb8c94aa5a72447f67eeaa6181cf3b2978f2f247a6629a1f09a91a`	`./Fair3.sol`

The latest version was made available in the course of the review:

FIELD	VALUE
Deployed Address	https://bscscan.com/address/0x6952c5408b9822295ba4a7e694d0c5ffdb8fe320
Tx Deploy	https://bscscan.com/tx/0xaa316939d8bd12e396e5f4cc00ce32c685f74d572a9b9947a2831cecc6dfbc7e
Deployer	https://bscscan.com/address/0x34940fcb935eb7cbaf96e0285cc816de670453a7
Block Number	47799935

Table 1. Fair Token's deployed properties

1.2. Audit Methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that were considered during the audit of the smart contract:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence

- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 2. Severity levels

1.3. Disclaimer

Fair and Free acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. Fair and Free understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, Fair and Free agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

1.4. Acceptance Minute

This final report served by Verichains to the Fair and Free will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the Fair and Free, the final report will be considered fully accepted by the Fair and Free without the signature.

2. AUDIT RESULT

2.1. Overview

The Fair Token was written in `Solidity` language, with the required version to be `^0.8.28`. The source code was written based on **OpenZeppelin's library**.

The token contract extends the `ERC20`, `Pausable`, and `Ownable` contracts. With `Ownable`, the contract deployer becomes the default contract owner. The token implementation includes several properties (as of this report's writing):

PROPERTY	VALUE
Name	Fair and Free
Symbol	FAIR[3]

Table 3. The Fair Token's properties

Additionally, the contract overrides several mechanisms from the default `ERC20` token. These include the pausable mechanism, whitelist mechanism, and transfer functions that use these two mechanisms. The owner has exclusive rights to manage whitelist accounts, mint tokens, and control system pauses. When the system is paused, token transfers are restricted to whitelisted accounts only. However, the token owner can ask a whitelisted account to transfer tokens while the contract is paused.

2.2. Findings

During the audit process, the audit team had identified no vulnerable issue in the smart contract code.

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Mar 14, 2025*	Public Report	Verichains Lab

Table 4. Report versions history