



# SMART CONTRACT SECURITY ASSESSMENT

PREPARED FOR

VIFOX COIN



hello@interfi.network



interfinetwork



www.interfi.network

# FOR VIFOX COIN

Date of Report December 08, 2025

Code 0xEC69F1351b66902Cd5E79f0924A5Ad049F682540

Platform Binance Chain

Website <https://ViFoxCoin.com>

X <https://x.com/ViFoxCoin>

Language Solidity

Methodology Automated Review, Unit Tests, Manual Review

Auditor InterFi

● Disclaimer: Smart contracts deployed on blockchains are inherently exposed to potential exploits, vulnerabilities, and security risks. Blockchain and cryptographic technologies are emerging and carry ongoing uncertainties. Please review the full audit report for detailed insights into risk severity, vulnerabilities, and audit scope limitations.

● Centralization Warning: Centralized privileges—regardless of intent or access control—introduce elevated risks to contract security and user trust.

● KYC Advisory: The project lacks verified third-party KYC of its owners, team, or deployers. Without independent KYC, transparency and accountability are reduced, increasing the risk of fraud or rug pulls.

● Verification: Verify this report: <https://www.github.com/interfinetwork>

# TABLE OF CONTENTS

<b>FOR VIFOX COIN</b>	<b>1</b>
<b>TABLE OF CONTENTS</b>	<b>3</b>
<b>1. SUMMARY</b>	<b>4</b>
1.1 Summary of Findings	4
1.2 Resolution Status	4
1.3 System Overview	5
1.4 Files in Scope	5
1.5 Out-of-Scope Assumptions	6
<b>2. METHODOLOGY</b>	<b>7</b>
2.1 Audit Objectives	7
2.2 Methodologies	7
2.3 Risk Categorization	8
2.4 Resolution Status Definitions	9
<b>3. FINDINGS</b>	<b>10</b>
<b>4. CENTRALIZATION</b>	<b>15</b>
4.1 Privileged Functions	15
<b>5. DISCLAIMER</b>	<b>18</b>
5.1 Confidentiality	18
5.2 No Financial Advice	18
5.3 Technical Disclaimer	18
5.4 Timeliness & Accuracy	19
5.5 Third-Party Links	19
<b>6. ABOUT</b>	<b>20</b>
6.1 Connect with Us	20

# 1. SUMMARY

The audit resulted in the identification of issues across a range of severity levels, including logic flaws, access control oversights, and design inconsistencies. All high-impact findings were communicated to the development team with clear recommendations for remediation. Where applicable, the team has confirmed implementation of fixes or provided justifications for design choices.

## 1.1 Summary of Findings

Severity	Count
Critical	0
Major	0
Medium	1
Minor	2
Unknown	1
Centralization	2

## 1.2 Resolution Status

Status	Count
Fixed	5
Partially Fixed	0
Acknowledged	1
Pending	0

### 1.3 System Overview

This system is a decentralized protocol comprising a suite of smart contracts. These contracts collectively define the rules, permissions, and operational workflows for managing on-chain assets, executing user interactions, and enforcing protocol-level logic. Smart contracts in this context are self-executing code units that autonomously manage the state and behavior of digital assets based on predefined conditions.

The protocol utilizes these contracts to enable key functionalities such as:

- Ownership and access control enforcement
- Permission and role-based actions
- Data storage and updates
- Event logging and auditability
- Batch processing and collection management

### 1.4 Files in Scope

InterFi was engaged by ViFox Coin to perform a security audit of the smart contracts. The audit scope was strictly limited to the files explicitly listed under the “Files in Scope” section. No other files or components were reviewed unless otherwise stated.

File	Path	Notes
VFX	<code>ViFoxCoin.sol</code>	Custom Module

## 1.5 Out-of-Scope Assumptions

The following components and assumptions were explicitly excluded from this audit:

- Frontend or backend integration logic.
- Off-chain components, scripts, or oracles.
- External contracts or libraries unless explicitly stated.
- Compiler-level or EVM-specific behavior outside the contract's scope.
- Governance or tokenomics-related decisions not implemented in code.
- All third-party dependencies as discussed in findings.

## 2. METHODOLOGY

### 2.1 Audit Objectives

This audit aims to ensure that the smart contract system is predictable, and behaves as intended under normal conditions. Primary audit objectives are to:

- Identify potential vulnerabilities or logic errors in the implementation.
- Evaluate adherence to best practices in smart contract development.
- Assess the correctness of access controls and permission systems.
- Recommend remediations or enhancements for improved security and performance.

### 2.2 Methodologies

The audit follows a layered security approach using both automated tools and manual techniques. We review the contracts for functional correctness, exploitability, and adherence to smart contract best practices:

Type	Tools & Techniques
Manual Code Review	Line-by-line analysis to check logic, permissions, and edge cases
Automated Analysis	Tools like Slither, MythX, or custom linters to catch known patterns
Static Analysis	Identification of bugs without executing the code (compile-time checks)
Unit Test Inspection	Evaluation of existing test coverage, assumptions, and potential false positives/negatives (if applicable)
Architecture Review	Mapping of privileged roles, callable paths, and contract interdependencies (if applicable)

## 2.3 Risk Categorization

Each issue identified during the audit is assigned a severity level based on its potential impact, exploitability, and likelihood of real-world abuse. These categories help prioritize remediation efforts:

Risk Severity	Definition
Critical	Represents a severe vulnerability that may result in complete contract compromise, such as asset theft, permanent loss of functionality, or unrestricted access. These issues are often easily exploitable and require immediate resolution.
Major	Indicates significant risk that can affect core contract behavior, enable unauthorized operations, or create unintended financial exposure. While not as urgent as critical risks, they should be remediated promptly.
Medium	These are moderate-level risks that may become exploitable under specific conditions. They often relate to logic errors, insufficient validation, or architectural oversights that could escalate over time.
Minor	Denotes issues that have low security impact but may degrade code quality, performance, or maintainability. These include inefficiencies, style violations, or redundant logic. Fixes are recommended for robustness.
Unknown	Risks where the severity cannot be confidently determined due to limited context, external dependencies, or ambiguous design intent. It is advised to treat these conservatively and address them proactively.
Centralization	Any function controlled by a single privileged role is treated as a critical risk, regardless of its purpose, due to the potential for misuse, override, or total asset control.

## 2.4 Resolution Status Definitions

All identified issues are also assigned a resolution status, indicating the current handling and response from the development team:

Status	Definition
● <span style="background-color: #a9f5d0; border-radius: 50%; padding: 2px 5px;">Fixed</span>	The issue has been remediated and verified as resolved during the re-audit or final check.
● <span style="background-color: #e1ecff; border-radius: 50%; padding: 2px 5px;">Partially Fixed</span>	The issue has been partially mitigated, but remnants or related concerns may still exist. Further attention may be required.
● <span style="background-color: #ffd700; border-radius: 50%; padding: 2px 5px;">Acknowledged</span>	The development team has accepted the finding but opted not to implement a fix.
● <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Pending</span>	The issue remains unresolved at the time of publication. It poses a potential risk and should be addressed.

### 3. FINDINGS

01	Vesting logic can permanently lock tokens for allocation wallets
Severity	<span>Medium</span>
Affected	Vesting struct, vestings mapping, vestedAmount, transferableAmount, ViFoxCoin._update, stake
Description	<p>A vesting recipient who uses staking heavily or simply moves tokens around over time can end up with:</p> <ul style="list-style-type: none"><li>○ A non-zero balance</li><li>○ But zero <code>transferableAmount</code></li><li>○ Meaning their funds are effectively frozen in that wallet address</li></ul>
Recommendation	<p>Explicitly document that allocation wallets are treated as “non-standard” ERC-20 holders and may become non-transferable once they have distributed <code>totalAllocation</code> tokens cumulatively.</p> <p>Consider excluding staking operations from vesting release if the intention is to allow vesting participants to stake freely:</p>
Status	<span>Fixed</span>
Comment	<p>The vesting mechanism is exclusively designed for initial token allocations to founding partners and participating companies. These addresses are predetermined and controlled, and no external token transfers will be made to them during the project lifecycle. Additionally, these allocation wallets are not intended to participate in staking activities. Since these vested addresses are not intended to receive any additional tokens from third parties or engage in staking, the risk of accidentally locking tokens does not apply in our use case.</p>

**02****Front-running possibilities**

Severity	N/A
Affected	<code>transferVFX</code> , <code>stake</code> , <code>unstake</code>
	MEV impact is limited to which user gets minted first when close to the <code>dailyMintLimit</code> ; there is no safety violation.
Description	Users may attempt to time <code>unstake</code> before large mint events to achieve higher reward bands, but this is normal economic behavior and does not break invariants or allow theft.
Comment	No FR/MEV angle appears to lead to unauthorized access or loss-of-funds. Effects are limited to expected market/economic competition.
Status	N/A

**03****Ownership transfer does not transfer ultimate admin**

Severity

Minor

Affected

AccessControl, transferOwnership

Description

OwnershipTransferred is more of a primary operator change than a true transfer of ultimate admin power.

Recommendation

Decide clearly who should be the true ultimate admin

Status

 Fixed

Comment

The complete ownership and all admin roles have been transferred to a multi-signature wallet requiring 2 out of 3 signatures. The previous owner's roles have been fully revoked and this transition has been recorded on-chain. The multi-sig wallet now serves as the ultimate admin with full control over DEFAULT\_ADMIN\_ROLE, ADMIN\_ROLE, and all privileged functions. This setup ensures clear governance structure and eliminates any ambiguity regarding the true ultimate admin.

**04****Potential self-DoS via large number of stake lots**

Severity

Minor

Affected

stake, unstake, getTotalStaked

Staking design creates one Stake struct per call to stake.

unstake iterates over all Stake entries for a user, and getTotalStaked does

Description

A user that performs a very large number of small stakes can make unstake revert due to gas limits, essentially locking their own funds.

Recommendation

Encourage off-chain UX to limit the number of stakes per user (aggregate stakes)

Status

 Fixed

We have implemented a minimum staking threshold of 50 tokens at the UI level to prevent users from creating multiple small stakes. This measure significantly reduces the likelihood of gas-heavy unstake operations, as users interacting through our platform cannot create stakes below this limit.

Comment

While this restriction is enforced on the frontend, the majority of our users interact exclusively through the official website, making this mitigation effective for practical use cases.

**05**

## Third-party dependencies

Severity

**Unknown**

Affected

Whole contract

The following external modules & assumptions are critical; their correctness is assumed, not verified in this audit.

### OpenZeppelin

ERC20, IERC20, IERC20Metadata, ERC20Pausable, Pausable, AccessControl, IAccessControl, ERC165, IERC165, ReentrancyGuard, IERC20Errors and other 6093 error interfaces

Description

Risk: Incorrect or modified versions could invalidate EIP-712 signing, nonReentrant, or token safety guarantees.

### Environment Assumptions

- EVM compatible chain with correct implementation of Solidity 0.8.20 semantics (checked arithmetic, etc.).
- No modifications to standard *OpenZeppelin* libraries.
- No proxy/upgrade mechanism wrapping this implementation.

Because these components are out-of-scope, overall oracle and token behavior has severity is unknown.

Recommendation

Protocol's economic safety hinges on:

- A well-designed, manipulation-resistant interfaces.
- Use of standard, audited ERC-20 tokens.

Status

 Acknowledged

## 4. CENTRALIZATION

Centralization is one of the leading causes of smart contract-related asset losses. When a contract assigns critical powers to a privileged role—such as an `owner`, `admin`, or designated controller—the associated risk becomes elevated, especially if that role is tied to a single externally owned account (EOA). In many cases, privileged roles serve operational or safety functions, including:

- Emergency/Operational Controls
- Contract Configuration

### 4.1 Noteworthy Privileged Functions

Function / Capability	Role / Requirement
<code>pause()</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>unpause()</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>setDailyMintLimit(uint256 newLimit)</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>resetDailyMint()</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>defineOperator(address operator)</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>changeOperator(address oldOp, address newOp)</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>removeOperator(address operator)</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>transferOwnership(address newOwner)</code>	<code>onlyRole(ADMIN_ROLE)</code>
<code>registerUser(uint256, address)</code>	<code>onlyRole(MINTER_ROLE)</code>
<code>transferVFX(uint256, uint256, address)</code>	<code>onlyRole(MINTER_ROLE)</code>
Grant/revoke <code>ADMIN_ROLE</code>	<code>onlyRole(DEFAULT_ADMIN_ROLE)</code>
Grant/revoke <code>MINTER_ROLE</code> (via admin)	<code>onlyRole(ADMIN_ROLE)</code>
Grant/revoke <code>DEFAULT_ADMIN_ROLE</code>	<code>onlyRole(DEFAULT_ADMIN_ROLE)</code>

01

## Centralized Privileges

Severity

Centralization

Description

EOAs with control can be compromised via phishing, private key leakage, or insider threats. Malicious or negligent use of privileges can lead to - token supply manipulation, disruption of trading via pausing, arbitrary fee changes or wallet exclusions, asset seizures or rerouting, etc.

### 4.1 Privileged Functions

Using Multi-Signature Wallets: Assign privileged roles to a multi-sig contract requiring signatures from multiple trusted parties. This reduces the impact of any single compromised key.

Time-Locked Functions: Introduce delays before executing sensitive operations, allowing time for community review or cancellation.

Recommendation

Role Revocation or Transfer: If privileges are no longer needed post-deployment, renounce them or migrate them to DAO governance.

Secure Key Management: Any private keys associated with privileged roles must be protected using hardware wallets, secret sharing schemes, or offline signing protocols.

Status

 Fixed

Comment

The complete ownership and all admin roles have been transferred to a multi-signature wallet requiring 2 out of 3 signatures. The previous owner's roles have been fully revoked and this transition has been recorded on-chain. The multi-sig wallet now serves as the ultimate admin with full control over DEFAULT\_ADMIN\_ROLE, ADMIN\_ROLE, and all privileged functions.

**02****Owners & roles can significantly affect the contract**

Severity

**Centralization****DEFAULT\_ADMIN\_ROLE**

- Grant and revoke ADMIN\_ROLE (true admin of the system as currently implemented).
- Indirectly control MINTER\_ROLE via role administration.

**ADMIN\_ROLE**

- Pause the token (pause, unpause) - effectively freezing all transfers.
- Adjust `dailyMintLimit`, defining the emission rate up to the hard `maxSupply`.
- Manage minter operators (`defineOperator`, `changeOperator`, `removeOperator`).
- Call `transferOwnership` (operator change), though `DEFAULT_ADMIN_ROLE` remains ultimate authority.

Governance hardening:

Decide which role is the true owner (`DEFAULT_ADMIN_ROLE` vs `ADMIN_ROLE`) and align:

Recommendation

- Self-administer that role via `_setRoleAdmin(role, role)` and keep other roles strictly subordinated.
- Provide a clear, one-step `transferOwnership` that moves all relevant admin roles, or explicitly document otherwise.

Status

 Fixed

Comment

Fix is discussed in Centralization-01

## 5. DISCLAIMER

InterFi Network provides professional smart contract audits for blockchain-based codebases (commonly known as smart contracts). This audit assessed the reviewed contract(s) for common vulnerabilities, centralization risks, and logic flaws. However, no audit can guarantee the complete absence of bugs or vulnerabilities. This report does not constitute a security guarantee, endorsement, or assurance of business model soundness or legal compliance.

The review is limited strictly to the source code and its logic as provided, and does not extend to compiler behavior, off-chain components, or external integrations. Due to the evolving nature of blockchain technology and associated risks, users should understand that all materials, including this audit report, are provided strictly on an “as is”, “as available”, and “with all faults” basis.

### 5.1 Confidentiality

This report is confidential and intended solely for the client. It may not be disclosed, reproduced, or relied upon by third parties without prior written consent from InterFi Network. All terms, including confidentiality, liability limitations, and scope, are governed by the audit agreement.

### 5.2 No Financial Advice

This report is not financial, investment, tax, legal, or regulatory advice. It should not be relied upon for making investment decisions or assessing the value, viability, or safety of any token, product, or platform. No part of this document should be interpreted as an endorsement or recommendation. InterFi Network accepts no liability for any actions taken based on this report.

### 5.3 Technical Disclaimer

InterFi disclaims all warranties—express, implied, or statutory—including merchantability, fitness for a particular purpose, title, and non-infringement. We do not guarantee that the reviewed contracts are error-free, fully secure, or meet any specific requirements. Audit results may contain false positives or negatives, and findings are subject to the context and limitations of the review scope.

#### 5.4 Timeliness & Accuracy

Audit results reflect the state of the code at the time of review. InterFi makes no commitment to update findings after publication. We do not warrant the accuracy, completeness, or timeliness of information delivered via this report.

#### 5.5 Third-Party Links

This report may contain references or links to external websites and social media accounts. InterFi Network is not responsible for the content or operation of third-party platforms and assumes no liability for actions taken based on their content.

## 6. ABOUT

InterFi Network is a leading provider of intelligent blockchain solutions, offering secure, scalable, and production-ready smart contract services. Our team specializes in the development, testing, and auditing of smart contracts across a wide range of blockchain ecosystems.

We have delivered:

- 300+ smart contract systems developed
- 2,000+ smart contracts audited
- 500,000+ lines of code reviewed and analyzed

Our technical expertise spans multiple languages including:

- Solidity for EVM-compatible chains (Ethereum, BNB Chain, Polygon, Avalanche, Cronos, Fantom, Velas, Metis, and more)
- Move for next-generation platforms such as Sui and Aptos
- Rust for advanced ecosystems like Solana, Near, and Cosmos SDK-based chains

### 6.1 Connect with Us

InterFi Network is driven by a multidisciplinary team of engineers, developers, UI/UX specialists, and blockchain researchers. The core team consists of 3 senior members supported by 4+ expert contributors across code auditing, tooling, and protocol design.

- Website: [interfi.network](http://interfi.network)
- Email: [hello@interfi.network](mailto:hello@interfi.network)
- GitHub: [github.com/interfinetwork](https://github.com/interfinetwork)
- Telegram (Engineering): [@interfaudits](https://t.me/interfaudits)
- Telegram (Onboarding): [@interfisupport](https://t.me/interfisupport)

THANK YOU