InterFi NETWORK

SMART CONTRACT AUDIT

interfinetwork

hello@interfi.network

https://interfi.network

PREPARED FOR

STAKEDGAIN

INTRODUCTION

Auditing Firm	InterFi Network
Client Firm	StakedGain
Methodology	Automated Analysis, Manual Code Review
Language	Solidity
Contract	0x2D16DD3FA406401FA2229fCFD9E9bCE487DCE7d6
Blockchain	Base Explorer
Centralization	Limited Privileged Access
Commit	7752599c1c4dc1b1692a31c6fde66ee92e379d60
Website	https://www.stakedgain.com/
Telegram	https://t.me/staked_gain/
X (Twitter)	https://twitter.com/Staked_Gain/
Report Date	May 11, 2024

ℹ Verify the authenticity of this report on our website: https://www.github.com/interfinetwork

EXECUTIVE SUMMARY

InterFi has performed the automated and manual analysis of solidity codes. Solidity codes were reviewed for common contract vulnerabilities and centralized exploits. Here's a quick audit summary:

Status	Critical 🔴	Major 🟠	Medium 🟡	Minor 🟢	Unknown 🟤
Open	0	1	1	0	0
Acknowledged	0	0	1	1	1
Resolved	0	0	0	2	0

ℹ️ Please note that smart contracts deployed on blockchains aren't resistant to exploits, vulnerabilities and/or hacks. Blockchain and cryptography assets utilize new and emerging technologies. These technologies present a high level of ongoing risks. For a detailed understanding of risk severity, source code vulnerability, and audit limitations, kindly review the audit report thoroughly.

ℹ️ Please note that centralization privileges regardless of their inherited risk status - constitute an elevated impact on smart contract safety and security.

ℹ️ Please note that our scope of work is strictly limited to reviewing the token smart contract. Token contract is straight-forward, with standard ERC20 features only. Liquidity is not added yet. Technically, the token contract is not yet initialized, hence, token audit is not useful in this situation yet.

⚠️ Please note, we have not conducted audits on any additional smart contracts, including but not limited to presale contracts, staking contracts, or any other contracts.

TABLE OF CONTENTS

TABLE OF CONTENTS 4

SCOPE OF WORK 5

AUDIT METHODOLOGY 6

RISK CATEGORIES 8

AUTOMATED ANALYSIS 9

INHERITANCE GRAPH 11

MANUAL REVIEW 12

DISCLAIMERS 20

ABOUT INTERFI NETWORK 23

SCOPE OF WORK

InterFi was consulted by StakedGain to conduct the smart contract audit of their solidity source codes.

The audit scope of work is strictly limited to mentioned solidity file(s) only:

- StakedGain.sol

ℹ️ If source codes are not deployed on the main net, they can be modified or altered before main-net deployment. Verify the contract's deployment status below:

Public Contract Link	
https://basescan.org/address/0x2d16dd3fa406401fa2229fcfd9e9bce487dce7d6#code	
Contract Name	StakedGain
Compiler Version	0.8.17
License	MIT

AUDIT METHODOLOGY

Smart contract audits are conducted using a set of standards and procedures. Mutual collaboration is essential to performing an effective smart contract audit. Here's a brief overview of InterFi's auditing process and methodology:

CONNECT

- The onboarding team gathers source codes, and specifications to make sure we understand the size, and scope of the smart contract audit.

AUDIT

- Automated analysis is performed to identify common contract vulnerabilities. We may use the following third-party frameworks and dependencies to perform the automated analysis:
  - Remix IDE Developer Tool
  - Open Zeppelin Code Analyzer
  - SWC Vulnerabilities Registry
  - DEX Dependencies, e.g., Pancakeswap, Uniswap
- Simulations are performed to identify centralized exploits causing contract and/or trade locks.
- A manual line-by-line analysis is performed to identify contract issues and centralized privileges. We may inspect below mentioned common contract vulnerabilities, and centralized exploits:

Centralized Exploits	○ Token Supply Manipulation ○ Access Control and Authorization ○ Assets Manipulation ○ Ownership Control ○ Liquidity Access ○ Stop and Pause Trading ○ Ownable Library Verification

Common Contract Vulnerabilities	○ Integer Overflow ○ Lack of Arbitrary limits ○ Incorrect Inheritance Order ○ Typographical Errors ○ Requirement Violation ○ Gas Optimization ○ Coding Style Violations ○ Re-entrancy ○ Third-Party Dependencies ○ Potential Sandwich Attacks ○ Irrelevant Codes ○ Divide before multiply ○ Conformance to Solidity Naming Guides ○ Compiler Specific Warnings ○ Language Specific Warnings

REPORT

- The auditing team provides a preliminary report specifying all the checks which have been performed and the findings thereof.
- The client's development team reviews the report and makes amendments to solidity codes.
- The auditing team provides the final comprehensive report with open and unresolved issues.

PUBLISH

- The client may use the audit report internally or disclose it publicly.

ℹ️ It is important to note that there is no pass or fail in the audit, it is recommended to view the audit as an unbiased assessment of the safety of solidity codes.

RISK CATEGORIES

Smart contracts are generally designed to hold, approve, and transfer tokens. This makes them very tempting attack targets. A successful external attack may allow the external attacker to directly exploit. A successful centralization-related exploit may allow the privileged role to directly exploit. All risks which are identified in the audit report are categorized here for the reader to review:

Risk Type	Definition
Critical 🔴	These risks could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
Major 🟠	These risks are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to high-risk severity.
Medium 🟡	These risks should be fixed, as they carry an inherent risk of future exploits, and hacks which may or may not impact the smart contract execution. Low-risk re-entrancy-related vulnerabilities should be fixed to deter exploits.
Minor 🟢	These risks do not pose a considerable risk to the contract or those who interact with it. They are code-style violations and deviations from standard practices. They should be highlighted and fixed nonetheless.
Unknown 🟤	These risks pose uncertain severity to the contract or those who interact with it. They should be fixed immediately to mitigate the risk uncertainty.

All statuses which are identified in the audit report are categorized here for the reader to review:

Status Type	Definition
Open	Risks are open.
Acknowledged	Risks are acknowledged, but not fixed.
Resolved	Risks are acknowledged and fixed.

AUTOMATED ANALYSIS

Symbol	Definition
🛑	Function modifies state
💵	Function is payable
🔒	Function is internal
🔐	Function is private
!	Function is important

Context	Implementation			
└	_msgSender	Internal 🔒		
IERC20	Interface			
└	totalSupply	External !		NO!
└	balanceOf	External !		NO!
└	transfer	External !	🛑	NO!
└	allowance	External !		NO!
└	approve	External !	🛑	NO!
└	transferFrom	External !	🛑	NO!
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
Ownable	Implementation	Context		

| └ | <Constructor> | Public ❗ | 🛑 | NO❗ |

| └ | owner | Public ❗ | | NO❗ |

| └ | renounceOwnership | Public ❗ | 🛑 | onlyOwner |

||||||

| **StakedGain** | Implementation | Context, IERC20, Ownable |||

| └ | <Constructor> | Public ❗ | 🛑 | NO❗ |

| └ | name | Public ❗ | | NO❗ |

| └ | symbol | Public ❗ | | NO❗ |

| └ | decimals | Public ❗ | | NO❗ |

| └ | totalSupply | Public ❗ | | NO❗ |

| └ | balanceOf | Public ❗ | | NO❗ |

| └ | transfer | Public ❗ | 🛑 | NO❗ |

| └ | allowance | Public ❗ | | NO❗ |

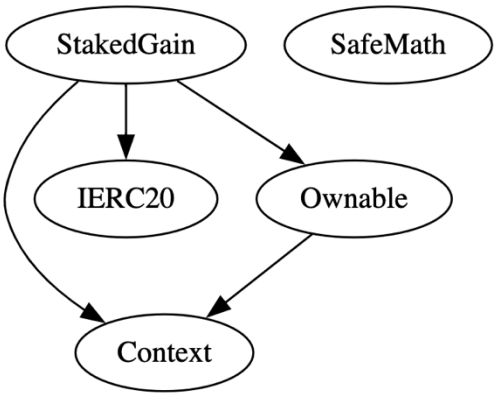
| └ | approve | Public ❗ | 🛑 | NO❗ |

| └ | transferFrom | Public ❗ | 🛑 | NO❗ |

| └ | _approve | Private 🔐 | 🛑 | |

| └ | _transfer | Internal 🔒 | 🛑 | |

INHERITANCE GRAPH



MANUAL REVIEW

Identifier	Definition	Severity
CEN-01	Centralized privileges	Minor 🟢

Important `onlyOwner` centralized privileges are listed below:

```
renounceOwnership()
```


RECOMMENDATION

Deployers', owners', administrators', and all other privileged roles' private-keys/access-keys/admin-keys should be secured carefully. These entities can have a single point of failure that compromises the security of the project. Manage centralized and privileged roles carefully, review PAGE 09 for more information.

RESOLUTION

StakedGain project uses minimal centralized privileges. Only standard ownable library is imported.

Identifier	Definition	Severity
CEN-02	Initial asset distribution	Medium 🟡

All of the initially minted assets are sent to the project owner when deploying the contract. This can be an issue as the project owner can distribute tokens without consulting the community.

```
uint256 private constant _tTotal = 100_000_000_000 * 10 ** _decimals;
```


RECOMMENDATION

Project must communicate with stakeholders and obtain the community consensus while distributing assets.

ACKNOWLEDGEMENT

StakedGain team acknowledged to distribute initially minted assets as specified in the pre-determined tokenomics.

Identifier	Definition	Severity
LOG-02	Potential front-running	Minor

Potential front-running also classified as – sandwich attack happens when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by front-running a transaction to purchase assets and make profits by back-running a transaction to sell assets.

RECOMMENDATION

Functions responsible for transferring assets – should be provided reasonable minimum output amounts, instead of zero.

ACKNOWLEDGEMENT

StakedGain smart contract doesn't collect any transaction tax. Front-running on EVM based chains in unavoidable.

Identifier	Definition	Severity
LOG-05	Possible race conditions	Major 🟠
LOG-06	Safe approval mechanism	

Allowance mechanism can lead to a race condition if not handled properly.

Fix:

```
function approve(address spender, uint256 amount) public override returns (bool) {
    _approve(_msgSender(), spender, amount);
    return true;
```

Implement safe approval mechanisms to prevent race conditions.

RECOMMENDATION

Set allowance to zero, before providing it a new value.

To prevent the ERC20 `approve`/`transferFrom` race condition, consider the use of `increaseAllowance` and `decreaseAllowance` functions.

Identifier	Definition
COD-09	Lack of contract balance withdraw

Smart contract may collect tokens, and ethers from external addresses. Some swap, and liquidity-add events may accumulate residual ethers, and tokens. Add `withdraw()` function to take out tokens and ethers from the contract.

Identifier	Definition	Severity
COD-10	Third Party Dependencies	Unknown

Smart contract is interacting with third party protocols e.g., Market Makers, Staking Contracts, External Token Contracts, *OpenZeppelin* tools. The scope of the audit treats third party entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised, and exploited. Moreover, upgrades in third parties can create severe impacts, e.g., increased transactional fees, deprecation of previous routers, etc.

RECOMMENDATION

Inspect third party dependencies regularly, and mitigate severe impacts whenever necessary.

ACKNOWLEDGEMENT

StakedGain team will inspect third party dependencies regularly, and push updates as required.

Identifier	Definition	Severity
COD-11	Inconsistent `SafeMath` usage	Medium 🟡
VOL-02	Redundant code in `SafeMath`	

Consolidate balance updates in `_transfer()` Function.

Fix:

```
_balances[from] = fromBalance.sub(amount);
_balances[to] = _balances[to].add(amount);
```

`SafeMath` is not required after solidity version 8.0, compiler performs built-in overflow, and underflow checks.

RECOMMENDATION

Consolidate balance calculation as recommended. Remove `SafeMath` if unnecessary.

Identifier	Definition	Severity
COM-01	Floating pragma	Minor

Smart contract uses floating pragma:

```
pragma solidity ^0.8.17;
```


RECOMMENDATION

Fix compiler pragma for better contract maintainability.

RESOLUTION

Smart contract is deployed with stable compiler version.

DISCLAIMERS

InterFi Network provides the easy-to-understand audit of solidity source codes (commonly known as smart contracts).

The smart contract for this particular audit was analyzed for common contract vulnerabilities, and centralization exploits. This audit report makes no statements or warranties on the security of the code. This audit report does not provide any warranty or guarantee regarding the absolute bug-free nature of the smart contract analyzed, nor do they provide any indication of the client's business, business model or legal compliance. This audit report does not extend to the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. Cryptographic tokens are emergent technologies, they carry high levels of technical risks and uncertainty. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. This audit report could include false positives, false negatives, and other unpredictable results.

CONFIDENTIALITY

This report is subject to the terms and conditions (including without limitations, description of services, confidentiality, disclaimer and limitation of liability) outlined in the scope of the audit provided to the client. This report should not be transmitted, disclosed, referred to, or relied upon by any individual for any purpose without InterFi Network's prior written consent.

NO FINANCIAL ADVICE

This audit report does not indicate the endorsement of any particular project or team, nor guarantees its security. No third party should rely on the reports in any way, including to make any decisions to buy or sell a product, service or any other asset. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. This audit report should not be used in any way

to make decisions around investment or involvement. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

FOR AVOIDANCE OF DOUBT, SERVICES, INCLUDING ANY ASSOCIATED AUDIT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

TECHNICAL DISCLAIMER

ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, INTERFI NETWORK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO SERVICES, AUDIT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, INTERFI NETWORK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, INTERFI NETWORK MAKES NO WARRANTY OF ANY KIND THAT ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CLIENT'S OR ANY OTHER INDIVIDUAL'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

TIMELINESS OF CONTENT

The content contained in this audit report is subject to change without any prior notice. InterFi Network does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following the publication.

LINKS TO OTHER WEBSITES

This audit report provides, through hypertext or other computer links, access to websites and social accounts operated by individuals other than InterFi Network. Such hyperlinks are provided for your reference and convenience only and are the exclusive responsibility of such websites' and social accounts' owners. You agree that InterFi Network is not responsible for the content or operation of such websites and social accounts and that InterFi Network shall have no liability to you or any other person or entity for the use of third-party websites and social accounts. You are solely responsible for determining the extent to which you may use any content at any other websites and social accounts to which you link from the report.

ABOUT INTERFI NETWORK

InterFi Network provides intelligent blockchain solutions. We provide solidity development, testing, and auditing services. We have developed 150+ solidity codes, audited 1000+ smart contracts, and analyzed 500,000+ code lines. We have worked on major public blockchains e.g., Ethereum, Binance, Cronos, Doge, Polygon, Avalanche, Metis, Fantom, Bitcoin Cash, Velas, Oasis, etc.

InterFi Network is built by engineers, developers, UI experts, and blockchain enthusiasts. Our team currently consists of 4 core members, and 6+ casual contributors.

Website: https://interfi.network

Email: hello@interfi.network

GitHub: https://github.com/interfinetwork

Telegram (Engineering): https://t.me/interfiaudits

Telegram (Onboarding): https://t.me/interfisupport

interfinetwork

hello@interfi.network

https://interfi.network

SMART CONTRACT AUDITS | SOLIDITY DEVELOPMENT AND TESTING

RELENTLESSLY SECURING PUBLIC AND PRIVATE BLOCKCHAINS