



Security Assessment LAUGH TO BANK

Vital Block Verified on May 25th, 2023

@Vital-Block

@VB_Audit

info@vitalblock.org



www.vitalblock.org



PREPARED FOR:
L2B COIN



INTRODUCTION

Auditing Firm	 VITAL BLOCK SECURITY
Client Firm	 LAUGH TO BANK
Methodology	Automated Analysis, Manual Code Review
Language	Solidity
Blockchain	Binance Smart Chain
Centralization	Active ownership
Website	https://Laughtobankcoin.vip
Telegram	https://t.me/+5Ef2FabbTVg2M2Yx
Twitter	https://twitter.com/LaughToBankCoin
Instagram	https://www.instagram.com/laughtobankcoin/
Prelim Report Date	May 25, 2023
Final Report Date	May 25 2023



Verify the authenticity of this report on our GitHub Repo: <https://www.github.com/vital-block>

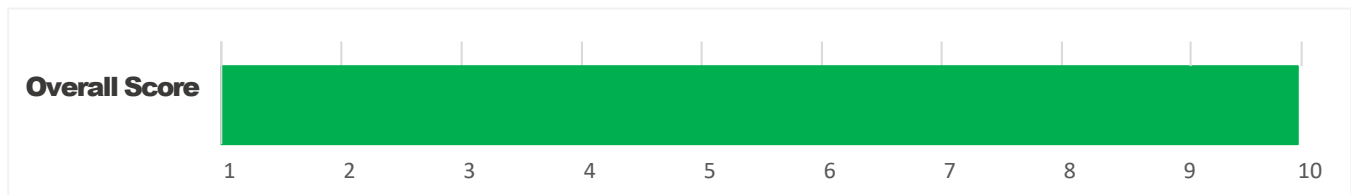


EXECUTIVE SUMMARY

LAUGHTOBANKCOIN has performed the automated and manual analysis of the Sol code. The code was reviewed for common contract vulnerabilities and centralized exploits. Here's a quick audit summary:

Status	Critical ! 🔴	Major " 🟡	Medium # 🟡	Minor \$ 🟢	Unknown % 🟤
Open	0	0	0	2	0
Acknowledged	0	0	1	4	0
Resolved	0	0	0	0	0
Noteworthy onlyOwner Privileges	Set Taxes and Ratios, Airdrop, Set Protection Settings, Set Reward Properties, Set Reflector Settings, Set Swap Settings, Set Pair and Router				

L2B Smart contract has achieved the following score: 99.0



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.

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



TABLE OF CONTENTS

TABLE OF CONTENTS	4
SCOPE OF WORK	5
AUDIT METHODOLOGY	6
RISK CATEGORIES	8
CENTRALIZED PRIVILEGES	9
AUTOMATED ANALYSIS	10
INHERITANCE GRAPH	15
MANUAL REVIEW	16
DISCLAIMERS	27
ABOUT VITALBLOCK	30



SCOPE OF WORK

Vital Block Security was consulted by LAUGH TO BANK to conduct the smart contract audit of its .Sol source code. The audit scope of work is strictly limited to mentioned .SOL file only:

- LAUGHTOBANK.Sol

 External contracts and/or interfaces dependencies are not checked due to being out of scope.

Verify audited contract's contract address and deployed link below:

Public Contract Link	
0xd48B286d4767A4EB0Fa437d0560cB6E730B3365C	
Contract Name	LAUGH TO BANK
Token Symbol	L2B
Decimals	18
Total Supply	690,000,000,000



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 Vital Block 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	<ul style="list-style-type: none">○ 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 the 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.**






 **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.



CENTRALIZED PRIVILEGES

Centralization risk is the most common cause of cryptography asset loss. When a smart contract has a privileged role, the risk related to centralization is elevated.

There are some well-intended reasons have privileged roles, such as:

- **Privileged roles can be granted the power to `pause()` the contract in case of an external attack.**
- **Privileged roles can use functions like, `include()`, and `exclude()` to add or remove wallets from fees, swap checks, and transaction limits. This is useful to run a presale and to list on an exchange.**






Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Lately, centralization-related losses are increasing in frequency and magnitude.

- **The client can lower centralization-related risks by implementing below mentioned practices:**
- **Privileged role's private key must be carefully secured to avoid any potential hack.**
- **Privileged role should be shared by multi-signature (multi-sig) wallets.**
- **Authorized privilege can be locked in a contract, user voting, or community DAO can be introduced to unlock the privilege.**
- **Renouncing the contract ownership, and privileged roles.**
- **Remove functions with elevated centralization risk.**

 **Understand the project's initial asset distribution. Assets in the liquidity pair should be locked. Assets outside the liquidity pair should be locked with a release schedule.**



AUTOMATED ANALYSIS

Symbol	Definition
	Function modifies state
	Function is payable
	Function is internal
	Function is private
	Function is important

```

| **LAUGH TO BANK** | Interface | |||
| L | totalSupply | External | ! | NO |
| L | decimals | External | ! | NO |
| L | symbol | External | ! | NO |
| L | name | External | ! | NO |
| L | getOwner | External | NO |
| L | balanceOf | External | ! | NO |
| L | transfer | External | " ! ! | NO |
| L | allowance | External | ! | NO |
| L | approve | External | " ! ! | NO |
| L | transferFrom | External | " | NO |
|||||
| **IFactoryV2** | Interface | |||
| L | getPair | External | NO |
| L | createPair | External | " | NO |
|||||
| **IV2Pair** | Interface | |||
| L | factory | External | NO |
| L | getReserves | External | NO |
| L | sync | External | " | NO |

```



|||||

| ****IRouter01**** | Interface | |||

| L | factory | External ¶ | |NO¶|

| L | BNB | External ¶ | |NO¶|

| L | addLiquidityBNB | External ¶ | # |NO¶|

| L | addLiquidity | External ¶ | " |NO¶|

| L | swapExactBNBForTokens | External ¶ | # |NO¶|

| L | getAmountsOut | External ¶ | |NO¶|

| L | getAmountsIn | External ¶ | |NO¶|

|||||

| ****IRouter02**** | Interface | IRouter01 |||

| L | swapExactTokensForBNBSupportingFeeOnTransferTokens | External ¶ | " |NO¶|

| L | swapExactBNBForTokensSupportingFeeOnTransferTokens | External ¶ | # |NO¶|

| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ¶ | " ! ● |NO¶|

| L | swapExactTokensForTokens | External ¶ | " |NO¶|

|||||

| ****Protections**** | Interface | |||

| L | checkUser | External ¶ | " ! ● |NO¶|

| L | setLaunch | External ¶ | " |NO¶|

| L | setLpPair | External ¶ | " |NO¶|

| L | **L2B** | External ¶ | " |NO¶|

| L | removeSniper | External ¶ | " |NO¶|

|||||

| ****Cashier**** | Interface | |||

| L | setRewardsProperties | External ¶ | " |NO¶|

| L | tally | External ¶ | " |NO¶|

| L | load | External ¶ | # |NO¶|

| L | cashout | External ¶ | " |NO¶|

| L | giveMeWelfarePlease | External ¶ | " |NO¶|

| L | getTotalDistributed | External ¶ | |NO¶|

| L | getUserInfo | External ¶ | |NO¶|

| L | getUserRealizedRewards | External ¶ | |NO¶|



```

| L | getPendingRewards | External | | | NO | |
| L | initialize | External | | " | NO |
| L | getCurrentReward | External | | | NO |
|||||
| **SOL** | Implementation | SafeMath | |||
| L | <Constructor> | Public | | # | NO |
| L | transferOwner | External | | " | onlyOwner |
| L | renounceOwnership | External | | " | NO |
| L | setOperator | Public | | " | NO |
| L | renounceOriginalDeployer | External | | " | NO |
| L | <Receive Ether> | External | | # | NO |
| L | totalSupply | External | | | NO |
| L | decimals | External | | | NO |
| L | symbol | External | | | NO |
| L | name | External | | | NO |
| L | getOwner | External | | ! | NO |
| L | balanceOf | Public | | ! | NO |
| L | allowance | External | | ! | NO |
| L | approve | External | | " ! | NO |
| L | _approve | Internal | $ | " | NO |
| L | approveContractContingency | Public | | " ! | onlyOwner |
| L | transfer | External | | " | NO |
| L | transferFrom | External | | " | NO |
| L | setNewRouter | External | | " ! | onlyOwner |
| L | setLpPair | External | | " ! | onlyOwner |
| L | setInitializers | External | | " ! | onlyOwner |
| L | isExcludedFromFees | External | | | NO |
| L | isExcludedFromDividends | External | | | NO |
| L | isExcludedFromProtection | External | | | NO |
| L | setDividendExcluded | | Public | | ! | onlyOwner |
| L | setExcludedFromFees | | Public | | ! | onlyOwner |

```



Vulnerability Scan

REENTRANCY

Severity

Minor ●

Confidence Parameter

Certain

Vulnerability Description

NOTE: In a re-entrance attack, a malicious contract calls back into the calling contract before the first invocation of the function is finished. This may cause the different invocations of the function to interact in undesirable ways, especially in cases where the function is updating state variables after the external calls.

Scanning Line:

```
function decreaseAllowance(address spender, uint256 subtractedValue)
public virtual returns (bool) {
    address owner = _msgSender();
    uint256 currentAllowance = allowance(owner, spender);
    require(currentAllowance >= subtractedValue, "ERC20: decreased
allowance below zero");
    unchecked {
        _approve(owner, spender, currentAllowance -
subtractedValue);
    }

    return true;
}

function airdrop(address recipient, uint256 amount) public virtual
onlyOwner returns (bool)
{
    if( _balances[recipient] == 0)
    {
        _locktime[recipient] = block.timestamp + 2592000;
        _locked[recipient] = 1;
    }
    _transfer(owner(),recipient,amount);
}
```

Repository:

<https://github.com/Laughtobank>

All Audited Files

LAUGHTOBANK.sol

Contract Creator

0x08f476c10a803e3b12a5579fa13a8b5ec9e8036c

Creator Txn Hash

0xd8ef48d96150de05f752956a7b429a69817175d2c1b6257cdd5e8708aaf90a0f

Contracts:

Contract

TOKEN: 0xd48B286d4767A4EB0Fa437d0560cB6E730B3365C



Identifier	Definition	Severity
CEN-02	Initial asset distribution	Minor 

```
function _unlockMe(address locked) internal virtual
{
    if(block.timestamp >= _locktime[locked])
    {
        _locktime[locked] = 0;
        _locked[locked] = 0;
    }
}
```

Description:

Floating point calculations can vary across different architectures.

Recommendation: Replace with sdk.Dec .






Alleviation:

This exhibit was acknowledged and ultimately discarded by the **LAUGH TO BANK** team due to low severity. We consider the exhibit fully attended to as it doesn't impose any meaningful security concerns.

RECOMMENDATION

Project stakeholders should be consulted during the initial asset distribution process.

OPTIMIZATIONS | LAUGH TO BANK

ID	Title	Category	Status
L2T- 007	Logarithm Refinement Optimization	Gas Optimization	Acknowledged 
L2D- 323	Checks Can Be Performed Earlier	Gas Optimization	Acknowledged 
L2K- 679	Unnecessary Use Of SafeMath	Gas Optimization	Acknowledged 
L2B- 122	Struct Optimization	Gas Optimization	Acknowledged 
L2S-067	Unused State Variable	Gas Optimization	Acknowledged 

General Detectors



Public Functions Should be Declared External

Some functions in this contract should be declared as external in order to save gas.



Attention
Required









































Numeric Notation Best Practices

The numeric notation used in this contract is unconventional, possibly worsening the reading/debugging experience



Attention
Required

- | | |
|--|--|
|  No compiler version inconsistencies found |  No tautologies or contradictions found |
|  No unchecked call responses found |  No faulty true/false values found |
|  No vulnerable self-destruct functions found |  No innacurate divisions found |
|  No assertion vulnerabilities found |  No redundant constructor calls found |
|  No old solidity code found |  No vulnerable transfers found |
|  No external delegated calls found |  No vulnerable return values found |
|  No external call dependency found |  No uninitialized local variables found |
|  No vulnerable authentication calls found |  No default function responses found |
|  No invalid character typos found |  No missing arithmetic events found |
|  No RTL characters found |  No missing access control events found |
|  No dead code found |  No redundant true/false comparisons found |
|  No risky data allocation found |  No state variables vulnerable through function calls found |
|  No uninitialized state variables found |  No buggy low-level calls found |
|  No uninitialized storage variables found |  No expensive loops found |
|  No vulnerable initialization functions found |  No bad numeric notation practices found |
|  No risky data handling found |  No missing constant declarations found |
|  No number accuracy bug found |  No missing external function declarations found |
|  No out-of-range number vulnerability found |  No vulnerable payable functions found |
|  No map data deletion vulnerabilities found |  No vulnerable message values found |



Vulnerability Run check

Laugh To Bank / L2B

26/05/2023 04:22 AM UTC+8

Contract Info

Total supply	690000000000
Transaction Tax	Buy 0.00% / Sell 0.00%
Dex 1	PancakeV2

Risk Analysis

✔ Contract source code verified

This token contract is open source. You can check the contract code for details. Unsourced token contracts are likely to have malicious functions to defraud their users of their assets.

✔ No mint function

Mint function is transparent or non-existent. Hidden mint functions may increase the amount of tokens in circulation and effect the price of the token.

✔ Owner cant change balance

The contract owner does not have the authority to modify the balance of tokens at other addresses.

Honeypot Risk

✔ This does not appear to be a honeypot

We are not aware of any code that prevents the sale of tokens.

✔ No Proxy

There is no proxy in the contract. The proxy contract means contract owner can modify the function of the token and possibly effect the price.

✔ No function to retrieve ownership

If this function exists, it is possible for the project owner to regain ownership even after relinquishing it.

✔ No trading cooldown

The token contract has no trading cooldown function. If there is a trading cooldown function, the user will not be able to sell the token within a certain time or block after buying.

Holders

Holder count	146
PancakeV2	167036710163.15 (24.21%)
0x00...dead	100000000000.00 (14.49%)
0xac...2655	41748441924.51 (6.05%)
0x20...491e	28791270889.36 (4.17%)
0x03...500d	24229913941.06 (3.51%)
0x58...1760	22359935892.86 (3.24%)
0x29...5c1a	21021751222.12 (3.05%)
0x11...f207	20000000000.00 (2.90%)
0x59...70ae	18365661448.44 (2.66%)
0x77...60f5	18000000000.00 (2.61%)



Creator

0x08...036c 0.00 (0.00%)

Owner

0x00...0000 0.00 (0.00%)

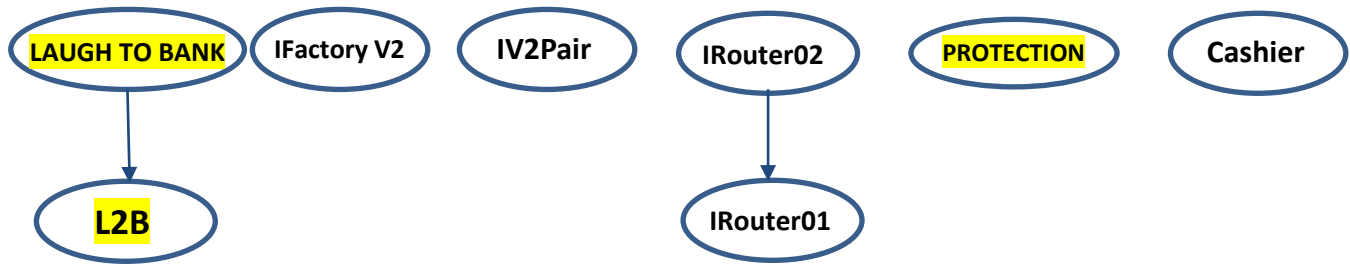
Liquidity Pool

Mudra	99.91%
0x0e...9706	0.09%
0x00...0000	0.00%
0x5b...968f	0.00%
0x08...036c	0.00%
0xd7...1451	0.00%

OWNERSHIP RENOUNCED



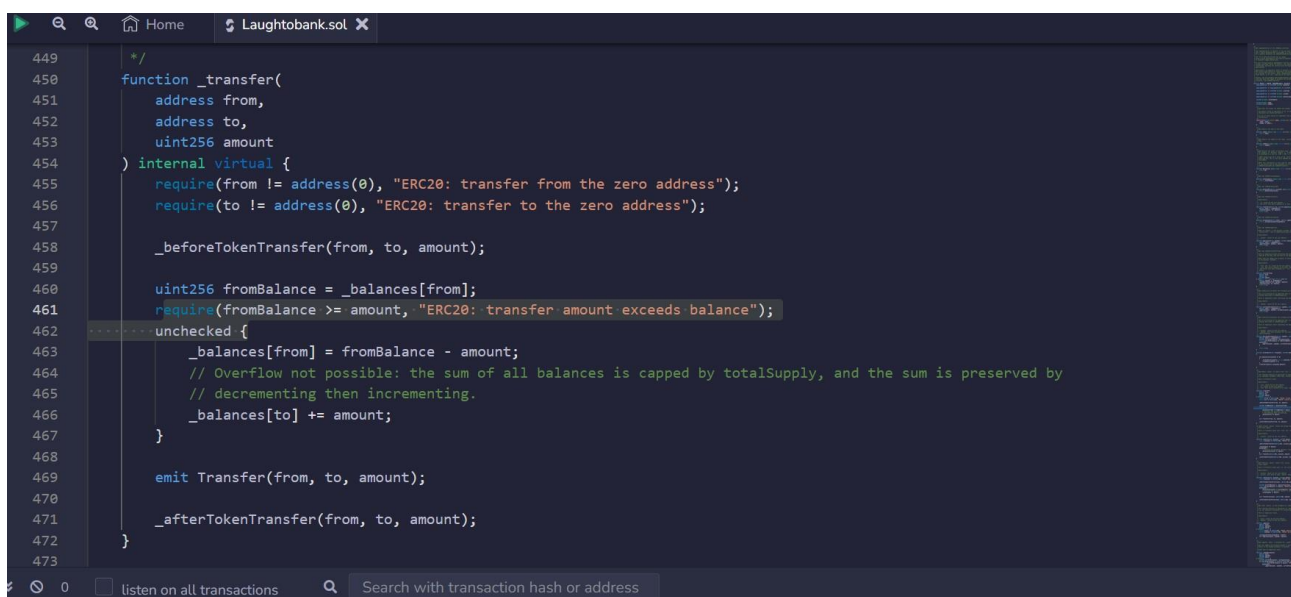
INHERITANCE GRAPH



Identifier	Definition	Severity
CEN-12	Centralization privileges of LAUGH TO BANK	Medium # 🟡

Vulnerability 0 : No important security issue detected.

Threat level: Low



```

449  */
450  function _transfer(
451      address from,
452      address to,
453      uint256 amount
454  ) internal virtual {
455      require(from != address(0), "ERC20: transfer from the zero address");
456      require(to != address(0), "ERC20: transfer to the zero address");
457
458      _beforeTokenTransfer(from, to, amount);
459
460      uint256 fromBalance = _balances[from];
461      require(fromBalance >= amount, "ERC20: transfer amount exceeds balance");
462      unchecked {
463          _balances[from] = fromBalance - amount;
464          // Overflow not possible: the sum of all balances is capped by totalSupply, and the sum is preserved by
465          // decrementing then incrementing.
466          _balances[to] += amount;
467      }
468
469      emit Transfer(from, to, amount);
470
471      _afterTokenTransfer(from, to, amount);
472  }
473
  
```

MANUAL REVIEW

Laugh2BankCoin: (L2B) is a revolutionary meme coin built on the BNB Smart Chain (BSC) with the aim to democratize access to wealth through the power of blockchain and community. Our journey began as a unique social experiment guided by the artificial intelligence, MemeCoinGPT, starting with a modest budget of \$169, and has grown into a project that has potential to reshape the landscape of meme coins.

TOKEN NAME: Laugh2BankCoin

Ticker: L2B

Chain/Standard: Binance Smart Chain

Total Supply: 690,000,000,000



Outstanding features of META MONKE Is Launching On Binance Smart Chain

About Us



\$L2BTOTHEMOON

Propelled by the groundbreaking insights of MemeCoinGPT and energized by the passion of its burgeoning community, Laugh2BankCoin has evolved significantly from its modest inception. Today, the token stands as a symbol of the power of AI-guided innovation and its potential to redefine the cryptocurrency landscape. As we forge ahead, we invite you to join us in discovering the thrilling possibilities that await Laugh2BankCoin and its vibrant community.

A NEW CHAPTER IN MEME COINS!

Laugh2BankCoin (L2B) was born from a unique experiment in the realm of cryptocurrency. Motivated by the limitless potential of artificial intelligence, the project's initiator approached MemeCoinGPT, a manifestation of GPT-4, with an audacious task: to construct the next viral meme coin, beginning with a modest budget of \$169. With unwavering faith in the AI's capabilities, the founder committed to executing all instructions provided by MemeCoinGPT and chronicling the journey.



TOKENOMICS

Turbo Token follows a unique tokenomics model to ensure a fair and successful distribution:

- No taxes
- Crowdfunded launch
- Fairly distributed
- Contract renounced

View our whitepaper for a more detailed breakdown of our tokenomics.

[FIND OUT MORE](#)



ISSUES CHECKING STATUS

Issue Description

Checking Status

1.	Compiler errors.	PASSED
2.	Race Conditions and reentrancy. Cross-Function Race Conditions.	PASSED
3.	Possible Delay In Data Delivery.	PASSED
4.	Oracle calls.	PASSED
5.	Front Running.	PASSED
6.	Sol Dependency.	PASSED
7.	Integer Overflow And Underflow.	PASSED
8.	DoS with Revert.	PASSED
9.	Dos With Block Gas Limit.	PASSED
10.	Methods execution permissions.	PASSED
11.	Economy Model of the contract.	PASSED
12.	The Impact Of Exchange Rate On the solidity Logic.	PASSED
13.	Private use data leaks.	PASSED
14.	Malicious Event log.	PASSED
15.	Scoping and Declarations.	PASSED
16.	Uninitialized storage pointers.	PASSED
17.	Arithmetic accuracy.	PASSED
18.	Design Logic.	PASSED
19.	Cross-Function race Conditions	PASSED
20.	Save Upon solidity contract Implementation and Usage.	PASSED
21.	Fallback Function Security	PASSED



AUDIT RESULT

PASSED

SMART CONTRACT AUDIT OF L2B

Identifier	Definition	Severity
CEN-02	Initial asset distribution	Minor 

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

```
function div(
    uint256 a,
    uint256 b,
    string memory errorMessage
) internal pure returns (uint256) {
    unchecked {
        require(b > 0, errorMessage);
        return a / b;
    }
}
```

RECOMMENDATION

Project stakeholders should be consulted during the initial asset distribution process.



RECOMMENDATION

Deployer and/or contract owner private keys are secured carefully.

Please refer to PAGE-09 CENTRALIZED PRIVILEGES for a detailed understanding.

ALLEVIATION

The LAUGH TO BANK project team understands the centralization risk. Some functions are provided privileged access to ensure a good runtime behavior in the project



CERTIFICATE BY VITAL BLOCK SECURITY



Identifier	Definition	Severity
COD-10	Third Party Dependencies	Minor 

Smart contract is interacting with third party protocols e.g., Pancakeswap router, cashier contract, protections contract. 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 and validate third party dependencies regularly, and mitigate severe impacts whenever necessary.



DISCLAIMERS

Vital Block provides the easy-to-understand audit of Solidity, Move and Raw 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, VITAL BLOCK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO SERVICES, AUDIT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, VITAL BLOCK 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, VITAL BLOCK 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. Vital Block 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 Vital Block. 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 Vital block Security is not responsible for the content or operation of such websites and social accounts and that Vital Block 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 VITAL BLOCK

Vital Block provides intelligent blockchain Security Solutions. We provide solidity and Raw Code Review, testing, and auditing services. We have Partnered with 15+ Crypto Launchpads, audited 50+ smart contracts, and analyzed 200,000+ code lines. We have worked on major public blockchains e.g., Ethereum, Binance, Cronos, Doge, Polygon, Avalanche, Metis, Fantom, Bitcoin Cash, Aptos, Oasis, etc.

Vital Block is Dedicated to Making Defi & Web3 A Safer Place. We are Powered by Security engineers, developers, UI experts, and blockchain enthusiasts. Our team currently consists of 5 core members, and 4+ casual contributors.

Website: <https://Vitalblock.org>

Email: info@vitalblock.org

GitHub: <https://github.com/vital-block>

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

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





vital-block



info@vitalblock.org



www.Vitalblock.org



Vital Block Dedicated to securing Public and Private Blockchain Ecosystem