AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

LTF

DATED : 7 APR 23'

AUDIT SUMMARY

Project name – LTF

Date: 7 April, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	2
Acknowledged	0	0	0	0	0
Resolved		0	0	0	0

USED TOOLS

Tools:

1.Manual Review: The code has undergone a line-by-line review by the Ace team.

2.BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3.Slither: The code has undergone static analysis using Slither.

Token Information

Name : Let Trump Free

Symbol : LTF

Decimals: 9

Network: Binance smart chain

Token Type: BEP20

Token Address :
0xb657b1aC4C5d5b4111261AE654c6F66DB5Bca9b8

Owner:
0x0a02cc8161F97D0fcBFbF4A58FCCAe194064926B
(at time of audit)

Deployer:
0x0a02cc8161F97D0fcBFbF4A58FCCAe194064926B

Token Information

Fees:

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 10%

Fees Privilige: None

Ownership : Owner

Minting: No

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: Including in fees, exluding from fees, including in reflections, excluding from reflections

AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code isexercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

VULNERABILITY CHECKLIST

- Return values of low-level calls
- Private modifier
- Multiple Sends
- Using Suicide
- Gas Limitand Loops
- Address hardcoded
- Exception Disorder
- Using inline assembly
- Divide before multiply
- Missing Zero Address Validation
- Compiler version not fixed
- **Gasless Send**
- Using block.timestamp
- Re-entrancy
- Tautology or contradiction
- Timestamp Dependence
- Revert/require functions
- Use of tx.origin
- Integer overflow/underflow
- Dangerous strict equalities
- Using SHA3
- Using throw

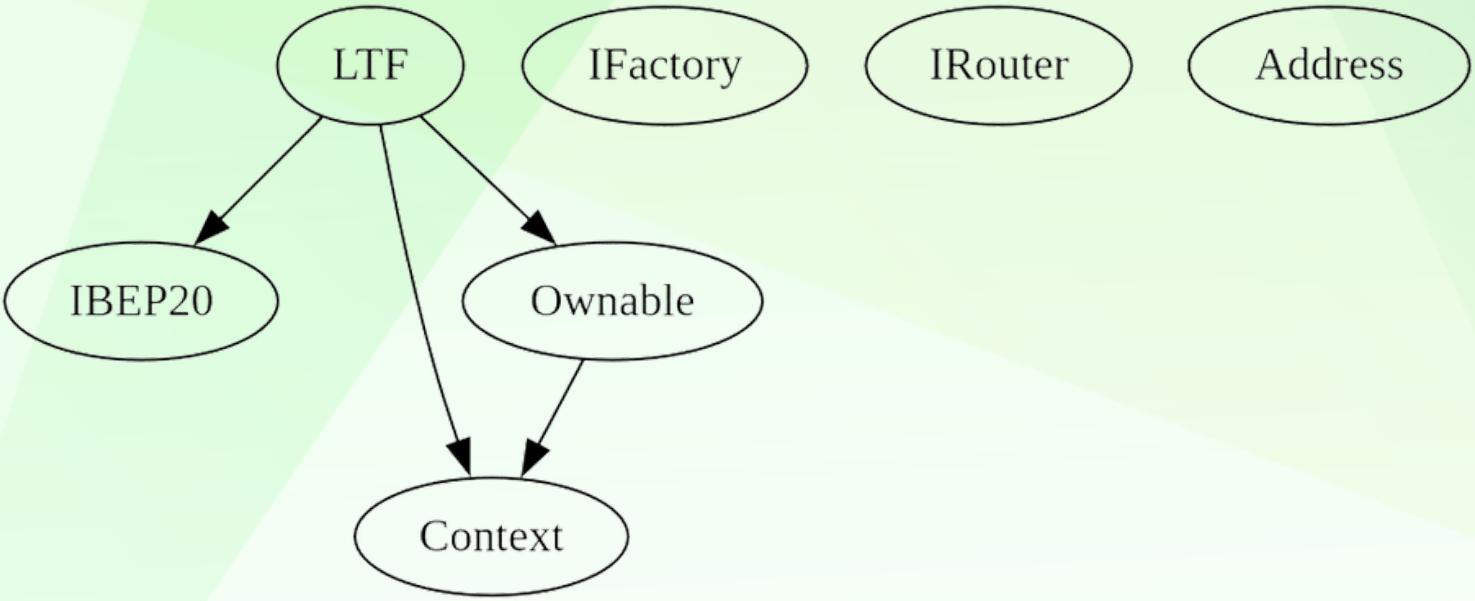
CLASSIFICATION OF RISK

Severity	Description
Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
Gas Optimization /Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
Critical	0
High-Risk	0
Medium-Risk	0
Low-Risk	0
Gas Optimization / Suggestions	2

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to modify fees (10% for each one)
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IBEP20	Interface				
└	totalSupply	External	|NO	
└	balanceOf	External	|NO	
└	transfer	External	|NO	
└	allowance	External	|NO	
└	approve	External	|NO	
└	transferFrom	External	|NO	
Context	Implementation				
└	_msgSender	Internal		
└	_msgData	Internal		
Ownable	Implementation	Context			
└	<Constructor>	Public	|NO	
└	owner	Public	|NO	
└	renounceOwnership	Public		onlyOwner
└	_setOwner	Private		
IFactory	Interface				
└	createPair	External	|NO	
IRouter	Interface				
└	factory	External	|NO	
└	WETH	External	|NO	
└	addLiquidityETH	External		
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External	|NO	
Address	Library				
└	sendValue	Internal		
LTF	Implementation	Context, IBEP20, Ownable			
└	<Constructor>	Public	|NO	
└	name	Public	|NO	
└	symbol	Public	|NO	
└	decimals	Public	|NO	
└	totalSupply	Public	|NO	
└	balanceOf	Public	|NO	
└	allowance	Public	|NO	
└	approve	Public	|NO	

CONTRACT ASSESMENT

└	transferFrom	Public		NO
└	increaseAllowance	Public		NO
└	decreaseAllowance	Public		NO
└	transfer	Public		NO
└	isExcludedFromReward	Public		NO
└	reflectionFromToken	Public		NO
└	tokenFromReflection	Public		NO
└	excludeFromReward	Public		onlyOwner
└	includeInReward	External		onlyOwner
└	excludeFromFee	Public		onlyOwner
└	includeInFee	Public		onlyOwner
└	setMarketingWallet	External		onlyOwner
└	setLiquidityWallet	External		onlyOwner
└	isExcludedFromFee	Public		NO
└	_reflectRfi	Private		
└	_takeMarketing	Private		
└	_takeLiquid	Private		
└	_getValues	Private		
└	_getTValues	Private		
└	_getRValues	Private		
└	_getRate	Private		
└	_getCurrentSupply	Private		
└	_approve	Private		
└	_transfer	Private		
└	_tokenTransfer	Private		
└	bulkExcludeFee	External		onlyOwner
└	<Receive Ether>	External	

Legend

Symbol	Meaning
	Function can modify state
	Function is payable

STATIC ANALYSIS

```
LTF.allowance(address,address).owner (contracts/Token.sol#236) shadows:
        - Ownable.owner() (contracts/Token.sol#59-61) (function)
LTF._approve(address,address,uint256).owner (contracts/Token.sol#479) shadows:
        - Ownable.owner() (contracts/Token.sol#59-61) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

LTF.constructor(address)._pair (contracts/Token.sol#192-195) lacks a zero-check on :
                - pair = _pair (contracts/Token.sol#198)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-zero-address-validation

LTF.includeInReward(address) (contracts/Token.sol#340-351) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#347)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Address.sendValue(address,uint256) (contracts/Token.sol#113-124) is never used and should be removed
Context._msgData() (contracts/Token.sol#41-44) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

LTF._rTotal (contracts/Token.sol#147) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#2) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#113-124):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#119)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#89) is not in mixedCase
Struct LTF.valuesFromGetValues (contracts/Token.sol#172-182) is not in CapWords
Constant LTF._decimals (contracts/Token.sol#143) is not in UPPER_CASE_WITH_UNDERSCORES
Constant LTF._name (contracts/Token.sol#153) is not in UPPER_CASE_WITH_UNDERSCORES
Constant LTF._symbol (contracts/Token.sol#154) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#42)" inContext (contracts/Token.sol#36-45)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

LTF._tTotal (contracts/Token.sol#146) should be constant
LTF.deadWallet (contracts/Token.sol#149) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

LTF.pair (contracts/Token.sol#141) should be immutable
LTF.router (contracts/Token.sol#140) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No issues found

FUNCTIONAL TESTING

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x1dffb67a7762068717da863e72b0e7df151589b99d2bd73d43a60511466a530a

2- Buying when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x76d8dd1f970a7bfed6c14dadf1bdddc427f6db9ca08bc9483b4a01f21be56934

3- Selling when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x98937f0bb967da35a4eb0bddf1c72369fe3cfdf7277ddd4dd5bb8ad9a0152e6f

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x1ca9bd431ddeb52bd96fe50328be923aba31ae95d27d083a5107039c9ce8d459

5- Buying when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0xf5c74ad279c488e1542bb2396ba2f2bd27a1361089c732e7c128712bf6b87d9d

6- Selling when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0x9dccbd6ba2da6b864b0e3082bfbcdd186e52b90fa909fccd3d314ac3ef117166

7- Transferring when not excluded from fees (10% tax) (passed):
https://testnet.bscscan.com/tx/0x55a1cc2e86da32a5a5a466645d5ef8ebc1d6816e9fa701b228b2c38a3a707fb7

FUNCTIONAL TESTING

8- Reflections (passed):

Marketing and liquidity wallets received tokens, holders received reflections

MANUAL TESTING

Informational – No event emission for fee transfers

Severity: Informational
Function: _reflectRfi - _takeLiquid - _takeMarketing
Lines: 375, 380, 389
Status: No Resolved
Fees are sent to marketing, liquidity wallets, but no events are emitted. This causes fee transfers to be hidden and hard to detect.
Recommendation:
emit a Transfer event from each function

MANUAL TESTING

Informational – Immutable tax

Severity: Informational
Function: ---
Lines: ---
Status: not resolved
Overview:
Contract has 10% tax for buy/sell/transfer actions. However this tax percentage is immutable, meaning it can not be changed later. A dynamic tax in a reasoable range is suggested as the market status is not predictable.

DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.

ABOUT AUDITACE

We specializes in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.

https://auditace.tech/

https://t.me/Audit_Ace

https://twitter.com/auditace_

https://github.com/Audit-Ace