CONTRACT WOLF

Blockchain Security - Smart Contract Audits

Security Assessment

May 17, 2022

Disclaimer

ContractWolf.io audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

ContractWolf does not provide any warranty on its released reports.

ContractWolf should not be used as a <u>decision</u> to invest into an audited project and is not affiliated nor partners to its audited contract projects.

ContractWolf provides transparent report to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within its **SMART CONTRACT**.

ContractWolf presence is to analyze, audit and assess the client's smart contract's code.

Each company or projects should be liable to its security flaws and functionalities.

Scope of Work

LESLARVERSE team agreed and provided us with the files that needs to be tested (Github, Bscscan, Etherscan, files, etc.). The scope of the audit is the main contract.

The goal of this engagement was to identify if there is a possibility of security flaws in the implementation of the contract or system.

ContractWolf will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, whitepaper and repository which has been provided by **LESLARVERSE.**

Network

Binance Smart Chain (BEP20)

Contract link

https://bscscan.com/address/0xC89a2C86D3656f27551080D33f59Beb92
0D0C80C

Website

http://www.leslarmetaverse.com

Telegram

https://t.me/leslarmetaverse_global

Twitter

http://www.twitter.com/leslarmetaverse

Instagram

https://instagram.com/leslarmetaverse?igshid=YmMyMTA2M2Y=

Discord

https://discord.com/invite/HRgUAjyJUk

Github

https://github.com/leslarmetaverse/LLVERSE-PUBLIC

Description

LESLARVERSE is a bridge that connects various crypto platforms such as Exchangers (example: Binance, PancakeSwap, Biswap), NFT marketplace platforms (example: OpenSea) and others.

Users can access multiple crypto platforms with just one platform, LESLARVERSE.

In LESLARVERSE there will also be educational and news platform facilities that make it easy for beginners to learn crypto and blockchain from the ground up. LESLARVERSE has a vision to become a platform that expands crypto adoption starting from the local market and then to the global market.

Logo

Risk Level Classification

Risk Level represents the classification or the probability that a certain function or threat that can exploit vulnerability and have an impact within the system or contract.
Risk Level is computed based on CVSS Version 3.0

Level	Value	Vulnerability
Critical	9 - 10	An Exposure that can affect the contract functions in several events that can risk and disrupt the contract
High	7 - 8.9	An Exposure that can affect the outcome when using the contract that can serve as an opening in manipulating the contract in an unwanted manner
Medium	4 - 6.9	An opening that could affect the outcome in executing the contract in a specific situation
Low	0.1 - 3.9	An opening but doesn't have an impact on the functionality of the contract
Informational	0	An opening that consists of information's but will not risk or affect the contract

Auditing Approach

Every line of code along with its functionalities will undergo manual review to check its security issues, quality, and contract scope of inheritance. The manual review will be done by our team that will document any issues that there were discovered.

Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:

- Review of the specifications, sources, and instructions provided to ContractWolf to make sure we understand the size, scope, and functionality of the smart contract.
- Manual review of code, our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities and security flaws.

2. Testing and automated analysis that includes:

- Testing the smart contract functions with common test cases and scenarios, to ensure that it returns the expected results.

3. Best practices review, the team will review the contract with the aim to improve efficiency, effectiveness, clarifications, maintainability, security, and control within the smart contract.

4. Recommendations to help the project take steps to secure the smart contract.

Used Code from other Frameworks/Smart Contracts (Direct Imports)

Imported Packages

- Buyback
- Collateralize
- Context
- Expensify
- Helpers
- IERC20
- IERC20Metadata
- IUniswapV2Factory
- IUniswapV2Pair
- IUniswapV2Router01
- IUniswapV2Router02
- LESLARVERSE
- Liquify
- Ownable
- Pancake
- RFI
- SafeMath
- Supply
- Tokenomics
- TxPolice

Description

Optimization enabled: Yes

Decimal: 9

Symbol: LLEVERSE

Max / Total supply: 1,000,000,000,000

Capabilities

Components

Version	Contracts	Libraries	Interfaces	Abstract
1.0	1	1	6	12

Exposed Functions

Version	Public	Private	External	Internal
1.0	18	12	112	52

State Variables

Version	Total	Public
1.0	62	32

Capabilities

Version	Solidity Versions Observed	Experimental Features	Can Receive Funds	Uses Assembly	Has Destroyable Contracts
1.0	v0.8.9		Yes	No	No

Inheritance Graph
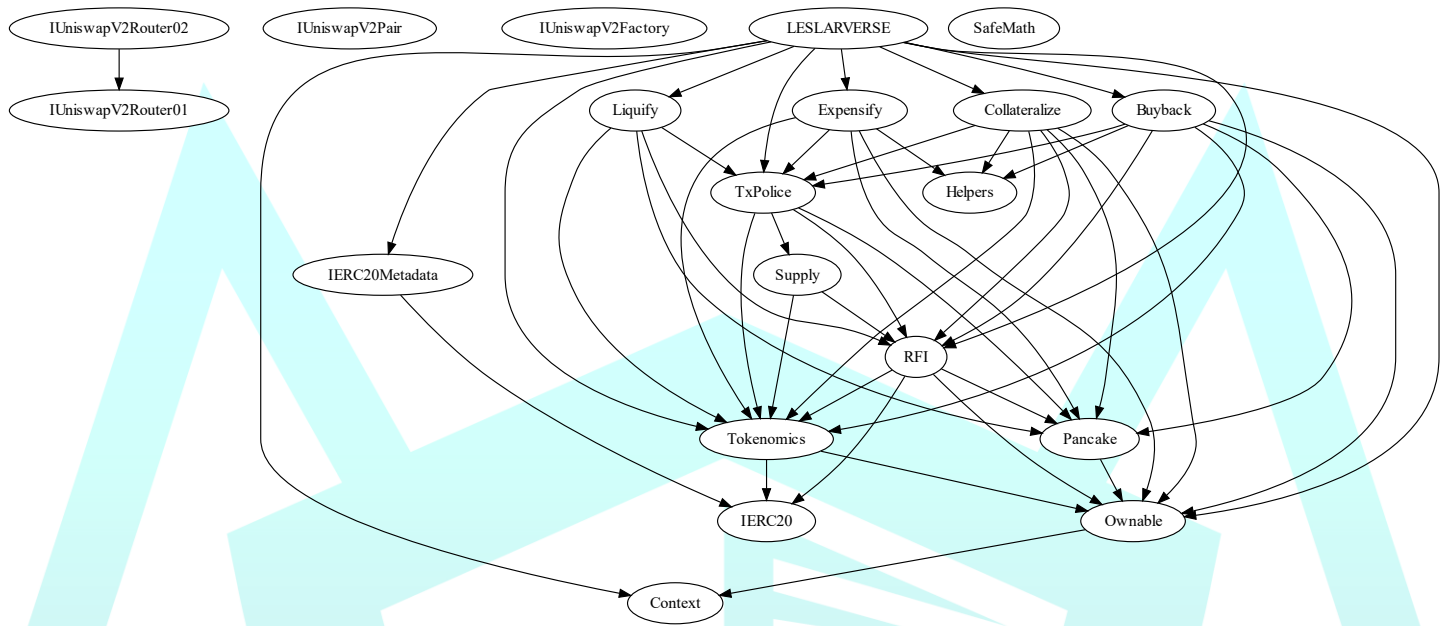
Correct implementation of Token Standard

Tested	Verified
✓	✓

Overall Checkup (Smart Contract Security)

Tested	Verified
✓	✓

Function	Description	Exist	Tested	Verified
TotalSupply	Information about the total coin or token supply	✓	✓	✓
BalanceOf	Details on the account balance from a specified address	✓	✓	✓
Transfer	An action that transfers a specified amount of coin or token to a specified address	✓	✓	✓
TransferFrom	An action that transfers a specified amount of coin or token from a specified address	✓	✓	✓
Approve	Provides permission to withdraw specified number of coin or token from a specified address	✓	✓	✓

Verify Claims

Statement	Exist	Tested	Deployer
Renounce Ownership	✓	✓	✓
Mint	✓	✓	✗
Burn	✓	✓	✗
Block	✓	✓	✓
Pause	—	—	—

Legend

Attribute	Symbol
Verified / Can	✓
Verified / Cannot	✗
Unverified / Not checked	🏳
Not Available	—

Write Functions of Contract

1. addAddressToLPs
2. addInitialLiquidity
3. approve
4. buyback
5. collateralize
6. decreaseAllowance
7. disableAllFeesTemporarily
8. disableSellLimit
9. enableSellLimit
10. increaseAllowance
11. initDEXRouter
12. pullBnbBuyback
13. removeAddressFromLPs
14. renounceOwnership
15. rescueBEP20Tokens
16. rescueBNB
17. restoreAllFees
18. setAntibot
19. setBuybackFee
20. setBuybackWallet
21. setCollateralFee
22. setCollateralWallet
23. setDevWallet
24. setLiquidityFee
25. setMarketingWallet
26. setMaxSellAllowanceMultiplier
27. setMaxTxAmountMultiplier
28. setMaxWalletSizeMultiplier
29. setMinToBuyback
30. setMinToCollateral
31. setMinToLiquify
32. setMinToTax
33. setProductDevWallet
34. setReflectionFee
35. setSellCycleHours
36. setTaxBuy
37. setTaxFee
38. setTaxSell
39. toggleLimitExemptions
40. toggleSpecialWallets
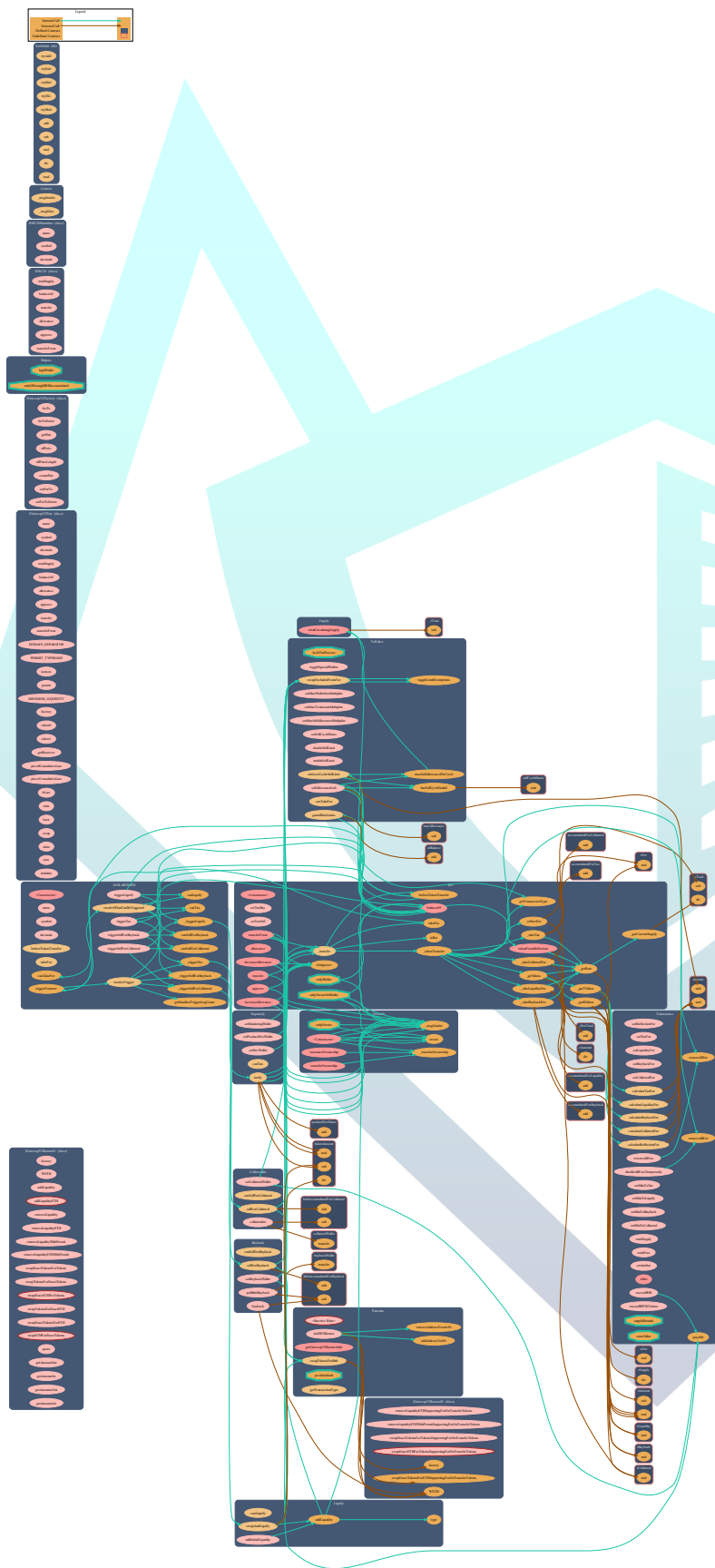41. transfer
42. transferFrom
43. transferOwnership
44. triggerLiquify
45. triggerSellForBuyback
46. triggerSellForCollateral
47. triggerTax

Call Graph

SWC Attacks

ID	Title	Status
SWC-136	Unencrypted Private Data On-Chain	PASSED
SWC-135	Code With No Effects	PASSED
SWC-134	Message call with hardcoded gas amount	PASSED
SWC-133	Hash Collisions with Multiple Variable Length Arguments	PASSED
SWC-132	Unexpected Ether balance	PASSED
SWC-131	Presence of unused variables	PASSED
SWC-130	Right-To Left Override control character (U+202E)	PASSED
SWC-129	Typographical Error	PASSED
SWC-128	DoS With Block Gas Limit	PASSED
SWC-127	Arbitrary Jump with Function Type Variable	PASSED
SWC-126	Insufficient Gas Griefing	PASSED
SWC-125	Incorrect Inheritance Order	PASSED
SWC-124	Write to Arbitrary Storage Location	PASSED
SWC-123	Requirement Violation	PASSED
SWC-122	Lack of Proper Signature Verification	PASSED
SWC-121	Missing Protection against Signature Replay Attacks	PASSED
SWC-120	Weak Sources of Randomness from Chain Attributes	PASSED
SWC-119	Shadowing State Variables	PASSED
SWC-118	Incorrect Constructor Name	PASSED
SWC-117	Signature Malleability	PASSED
SWC-116	Block values as a proxy for time	PASSED
SWC-115	Authorization through tx.origin	PASSED
SWC-114	Transaction Order Dependence	PASSED
SWC-113	DoS with Failed Call	PASSED
SWC-112	Delegate call to Untrusted Callee	PASSED
SWC-111	Use of Deprecated Solidity Functions	PASSED

SWC-110	Assert Violation	PASSED
SWC-109	Uninitialized Storage Pointer	PASSED
SWC-108	State Variable Default Visibility	PASSED
SWC-107	Reentrancy	PASSED
SWC-106	Unprotected SELFDESTRUCT Instruction	PASSED
SWC-105	Unprotected Ether Withdrawal	PASSED
SWC-104	Unchecked Call Return Value	PASSED
SWC-103	Floating Pragma	LOW ISSUE
SWC-102	Outdated Compiler Version	PASSED
SWC-101	Integer Overflow and Underflow	PASSED
SWC-100	Function Default Visibility	PASSED

AUDIT PASSED

Low Issues

A floating pragma is set (SWC-103)	L: 1	IUniswapV2Factory.sol IUniswapV2Pair.sol IUniswapV2Router01.sol
	L: 2	LESLARVERSE.sol RFI.sol Liquify.sol Expensify.sol Buyback.sol Collateralize.sol TxPolice.sol Pancake.sol Helpers.sol Supply.sol
	L: 3	IUniswapV2Router02.sol
	L: 4	SafeMath.sol Context.sol Ownable.sol IERC20.sol IERC20Metadata.sol Tokenomics.sol

Audit Comments

- Deployer can renounce ownership
- Deployer can transfer ownership
- Deployer can initialize DEX router
- Deployer can add/remove address to liquidity pool
- Deployer can set buy/sell fee with an indefinite amount
- Deployer can set reflection fee with an indefinite amount
- Deployer can set tax fee with an indefinite amount
- Deployer can set liquidity fee with an indefinite amount
- Deployer can set buyback fee with an indefinite amount
- Deployer can set collateral fee with an indefinite amount
- Deployer can disable/restore all fees
- Deployer can set minimum token to tax
- Deployer can set minimum token to liquify
- Deployer can set minimum token to buyback
- Deployer can set minimum token to collateral
- Deployer can block/unblock address for anti-bot
- Deployer can collect BNB from contract
- Deployer can toggle limit exemptions
- Deployer can add/remove special wallet address
- Deployer can set max wallet size multiplier with an amount greater than 0 and less than 1000000
- Deployer can set max transaction amount multiplier with an amount greater than 0 and less than 1000000
- Deployer can set max sell allowance multiplier with an amount greater than 0
- Deployer can set cycle hours with an amount greater than 0

- Deployer can enable/disable sell limit
- Deployer can set collateral wallet address
- Deployer can set buyback wallet address
- Deployer can set product dev wallet address
- Deployer can set dev wallet address
- Deployer can set marketing wallet address
- Deployer can trigger liquify
- Deployer can set trigger to sell for buyback
- Deployer can set trigger to sell for collateral
- Deployer can trigger tax
- Deployer can collateralize BNB
- Deployer can buyback and send to dead address
- Deployer can transfer BNB to buyback wallet
- Deployer can add liquidity
- Deployer can collect tokens from contract
- Deployer cannot lock contract
- Deployer cannot mint after initial deployment
- Deployer cannot burn tokens

CONTRACTWOLF

Blockchain Security - Smart Contract Audits