SMART CONTRACT SECURITY ASSESSMENT

PROJECT:

LUNASPHERE

DATE:

03 MAY, 2023

https://t.me/SafuAudit

www.safuaudit.com

Introduction

Client	Lunasphere
Language	Solidity
Contract Address	0x897504Dc693B30c367cdaCfA88578582686C1d95
Owner	0x2938a6f53d914ca944fAE2b19D659149188d5278
Deployer	0x2938a6f53d914ca944fAE2b19D659149188d5278
SHA-256 Hash	d670945901f84ee84ef53905f7cb71f53ac421e6
Decimals	18
Supply	10000000000000
Platform	Binance Smart Chain
Compiler	v0.8.4+commit.c7e474f2
Optimization	Yes with 200 runs
Website	https://lunaspheretoken.com/
Twitter	https://twitter.com/Lunaspheretoken
Telegram	https://t.me/lunasphere

Overview

Fees

- Buy fees: 10%
- Sell fees: 10%

Fees privileges

- Can't set fees above 25%

Ownership

- Owned

Minting

- No

Max Tx Amount

- Can't set max Tx

Pause

- Can't pause

Blacklist

- Can't blacklist

Other Privileges

- Can exclude from dividends
- Can exclude from fees

Table Of Contents

Risk Classification

Critical

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium

Issues on this level could potentially bring problems and should eventually be fixed.

Minor

Issues on this level are minor details and warning that can remain unfixed but would be better fixed at some point in the future

Informational

Information level is to offer suggestions for improvement of efficacity or security for features with a risk free factor.

Contract Inspection

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
IERC20Metadata	Interface	IERC20		
Context	Implementation			
ERC20	Implementation	Context, IERC20, IERC20Metadata		
Ownable	Implementation	Context		
SafeMath	Library			
Clones	Library			
Address	Library			
IUniswapV2Factory	Interface			
IUniswapV2Router01	Interface			
IUniswapV2Router02	Interface	IUniswapV2Router01		
IPinkAntiBot	Interface			
IERC20Upgradeable	Interface			
IERC20MetadataUpgradeable	Interface	IERC20Upgradeable		
Initializable	Implementation			
ContextUpgradeable	Implementation	Initializable		
ERC20Upgradeable	Implementation	Initializable, ContextUpgradeable, IERC20Upgradea		
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
IUniswapV2Pair	Interface			
SafeMathInt	Library			
SafeMathUint	Library			
IterableMapping	Library			
DividendPayingTokenInterface	Interface			
DividendPayingTokenOptionalInterface	Interface			
DividendPayingToken	Implementation	ERC20Upgradeable, OwnableUpgradeable, Dividenc		
BABYTOKENDividendTracker	Implementation	OwnableUpgradeable, DividendPayingToken		
BaseToken	Implementation			
AntiBotBABYTOKEN	Implementation	ERC20, Ownable, BaseToken		
└	<Constructor>	Public ❗	💸	ERC20
└	setEnableAntiBot	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💸	NO❗
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	excludeFromFees	External ❗	🛑	onlyOwner
└	excludeMultipleAccountsFromFees	External ❗	🛑	onlyOwner
└	setMarketingWallet	External ❗	🛑	onlyOwner

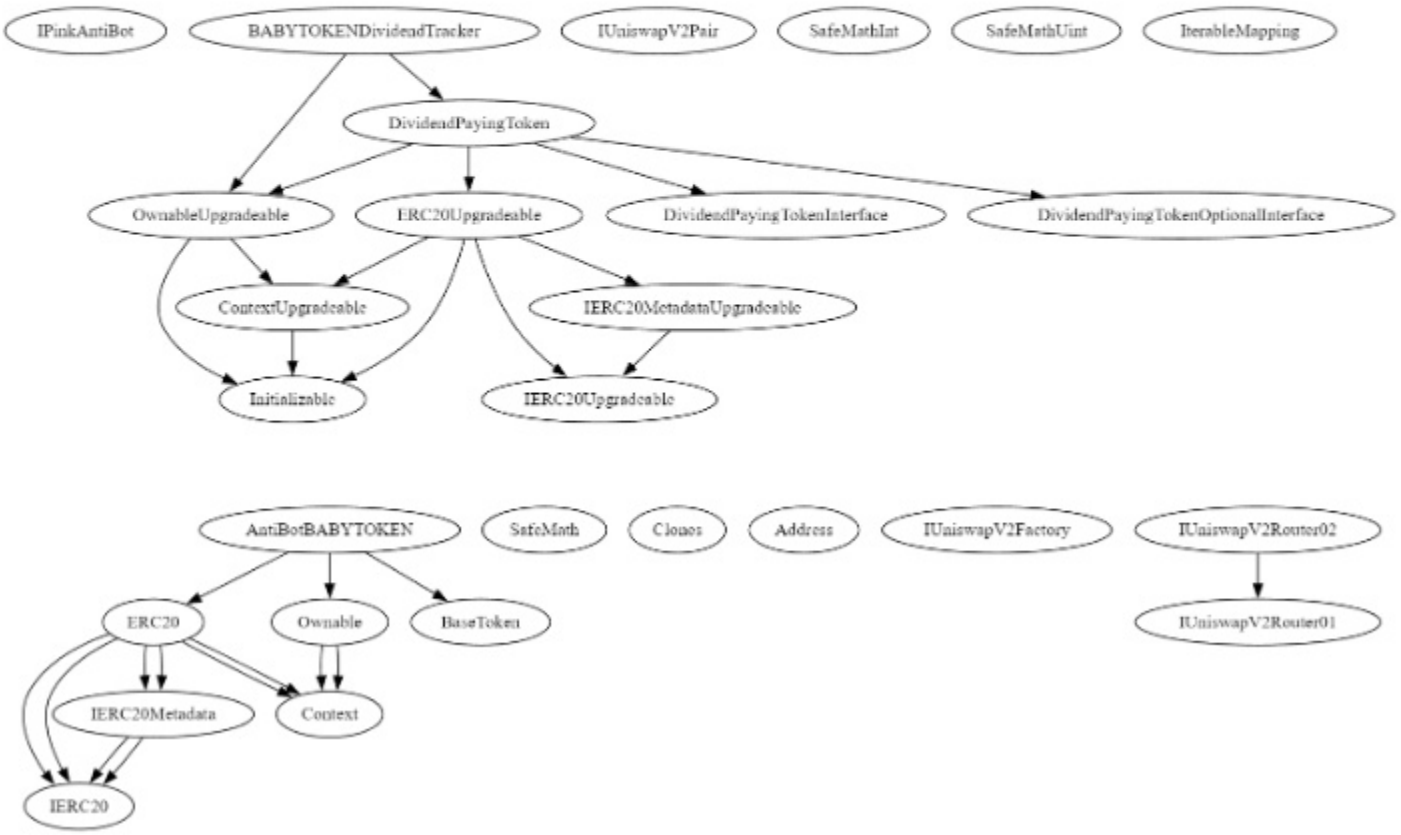
Contract Inspection

└	setTokenRewardsFee	External ❗	🛑	onlyOwner
└	setLiquiditFee	External ❗	🛑	onlyOwner
└	setMarketingFee	External ❗	🛑	onlyOwner
└	_setAutomatedMarketMakerPair	Private 🔐	🛑	
└	updateGasForProcessing	Public ❗	🛑	onlyOwner
└	updateClaimWait	External ❗	🛑	onlyOwner
└	getClaimWait	External ❗		NO❗
└	updateMinimumTokenBalanceForDividends	External ❗	🛑	onlyOwner
└	getMinimumTokenBalanceForDividends	External ❗		NO❗
└	getTotalDividendsDistributed	External ❗		NO❗
└	isExcludedFromFees	Public ❗		NO❗
└	withdrawableDividendOf	Public ❗		NO❗
└	dividendTokenBalanceOf	Public ❗		NO❗
└	excludeFromDividends	External ❗	🛑	onlyOwner
└	isExcludedFromDividends	Public ❗		NO❗
└	getAccountDividendsInfo	External ❗		NO❗
└	getAccountDividendsInfoAtIndex	External ❗		NO❗
└	processDividendTracker	External ❗	🛑	NO❗
└	claim	External ❗	🛑	NO❗
└	getLastProcessedIndex	External ❗		NO❗
└	getNumberOfDividendTokenHolders	External ❗		NO❗
└	_transfer	Internal 🔒	🛑	
└	swapAndSendToFee	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	
└	swapTokensForCake	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	swapAndSendDividends	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

Contract Inheritance



Inheritance is a feature of the object-oriented programming language. It is a way of extending the functionality of a program, used to separate the code, reduces the dependency, and increases the re-usability of the existing code. Solidity supports inheritance between smart contracts, where multiple contracts can be inherited into a single contract.

Vulnerabilities Test

Test Name	Result
Function Default Visibility	Passed
Integer Overflow and Underflow	Passed
Outdated Compiler Version	Passed
Floating Pragma	Passed
Unchecked Call Return Value	Passed
Unprotected Ether Withdrawal	Passed
Unprotected SELF-DESTRUCT Instruction	Passed
Reentrancy	Passed
State Variable Default Visibility	Passed
Uninitialized Storage Pointer	Passed
Assert Violation	Passed
Use of Deprecated Solidity Functions	Passed
Delegate Call to Untrusted Callee	Passed
DoS with Failed Call	Passed
Transaction Order Dependence	Passed
Authorization through tx.origin	Passed
Block values as a proxy for time	Passed
Signature Malleability	Passed
Incorrect Constructor Name	Passed

Vulnerabilities Test

Test Name	Result
Shadowing State Variables	Passed
Weak Sources of Randomness from Chain Attributes	Passed
Missing Protection against Signature Replay Attacks	Passed
Lack of Proper Signature Verification	Passed
Requirement Violation	Passed
Write to Arbitrary Storage Location	Passed
Incorrect Inheritance Order	Passed
Insufficient Gas Griefing	Passed
Arbitrary Jump with Function Type Variable	Passed
DoS With Block Gas Limit	Passed
Typographical Error	Passed
Right-To-Left-Override control character (U+202E)	Passed
Presence of unused variables	Passed
Unexpected Ether balance	Passed
Hash Collisions With Multiple Variable Length Arguments	Passed
Message call with the hardcoded gas amount	Passed
Code With No Effects	Passed
Unencrypted Private Data On-Chain	Passed

Findings

ID	Category	Issue	Severity
CE-OF	Centralization	Owner Accessible Functions	Optimization

CE-OF Owner Accessible Functions

Lines # multiple lines

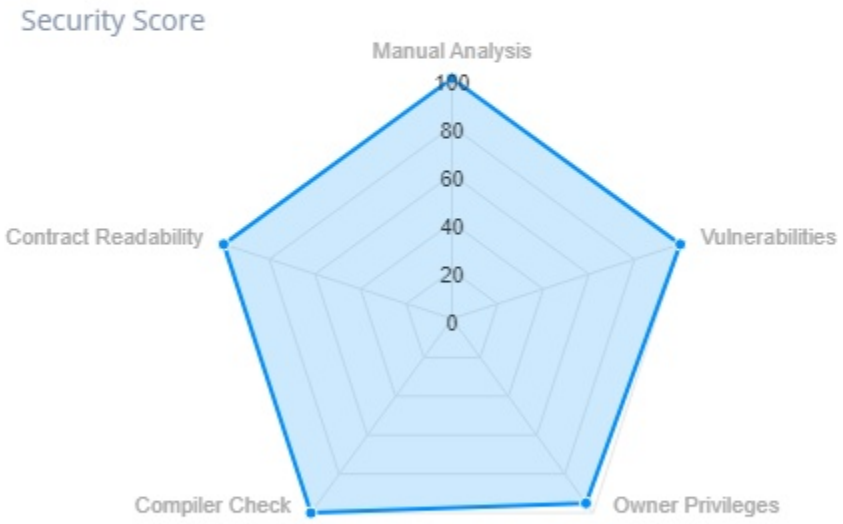
Description

The role OnlyOwner has authority over 22 functions that can manipulate the project functionality. Any compromise to the owner account may allow a hacker to take advantage of this authority.

Recommendation

We advise the client to carefully manage the privilege accounts' private key to avoid any potential risks of being hacked. Renounce Ownership at some point in time.

Security Score



Manual Analysis Score	100
Vulnerabilities Score	100
Contract Readability Score	100
Owner Privileges	95
Compiler Score	100
Total	**99**

Conclusion

Lunasphere contract uses ERC20 standard functionality with taxes for buy/sell and a reward system (dividends) for holders.

Disclaimer

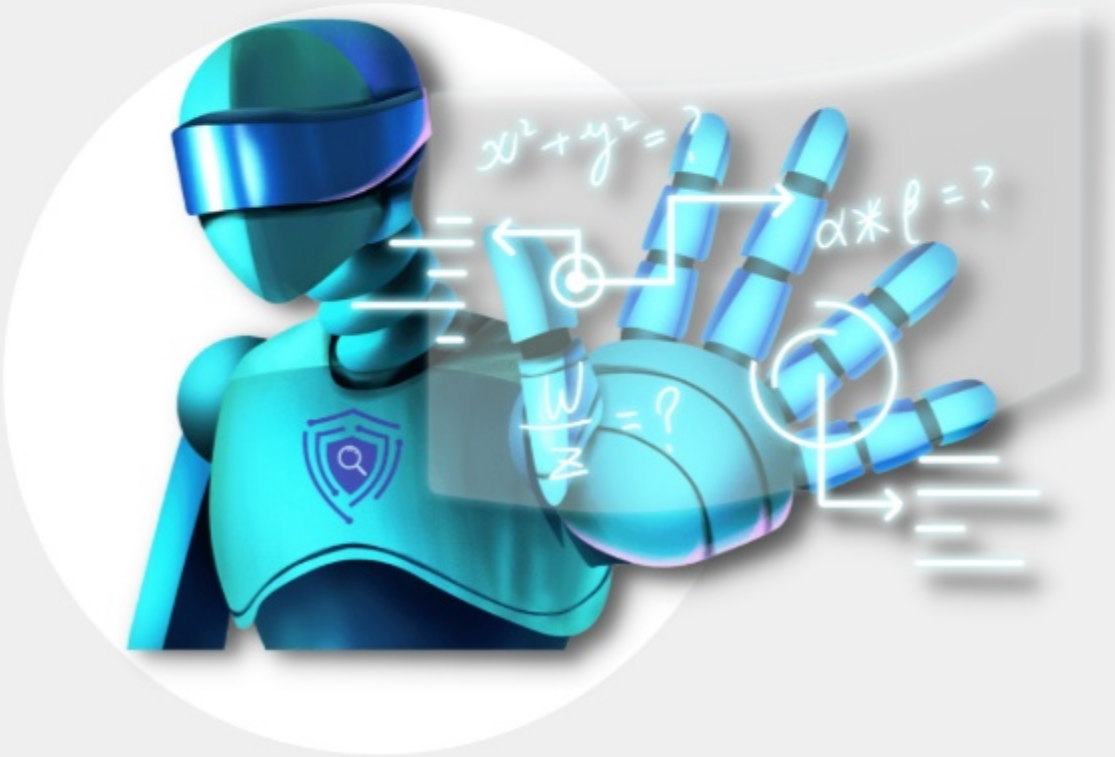
SafuAudit.com is not a financial institution and the information provided on this website does not constitute investment advice, financial advice, trading advice, or any other sort of advice. You should not treat any of the website's content as such. Investing in crypto assets carries a high level of risk and does not hold guarantees for not sustaining financial loss due to their volatility.

Accuracy of Information
SafuAudit will strive to ensure the accuracy of the information listed on this website although it will not hold any responsibility for any missing or wrong information. SafuAudit provides all information as is. You understand that you are using any and all information available here at your own risk. Any use or reliance on our content and services is solely at your own risk and discretion.

The purpose of the audit is to analyze the on-chain smart contract source code and to provide a basic overview of the project.

While we have used all the information available to us for this straightforward investigation, you should not rely on this report only — we recommend proceeding with several independent audits Be aware that smart contracts deployed on a blockchain aren't secured enough against external vulnerability or a hack. Be aware that active smart contract owner privileges constitute an elevated impact on the smart contract safety and security. Therefore, SafuAudit does not guarantee the explicit security of the audited smart contract. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

SAFUAUDIT
SMART CONTRACT AUDITS AND BLOCKCHAIN SECURITY



"Only in growth, reform, and change, paradoxically enough, is true security to be found."