Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

SHELBY Token

Deployer address

0xcaf28025150577f3bf8719f45399c4af2d88c94b

Client contacts:

SHELBY Token team

Blockchain

Binance Smart Chain

Project website:

https://www.shelbytoken.io

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

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.

Background

TechRate was commissioned by SHELBY Token to perform an audit of smart contracts:

https://bscscan.com/address/0xc95278Cd6e51bc2e1E30CF660E82819d296152D9#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 10.11.2021

Contract name	SHELBY Token
Contract address	0xc95278Cd6e51bc2e1E30CF660E82819d296152D9
Total supply	1,000,000,000
Token ticker	SBY
Decimals	18
Token holders	2,211
Transactions count	18,727
Top 100 holders dominance	86.51%
Dividend tracker	0x959e876d19e3f3bd55f262a16213a33a9f65cbd1
Total fees	15
Rewards fee	5
Uniswap V2 pair	0xa7cb9729c0152c8107f8821045b84b67c9c647ea
Contract deployer address	0xcaf28025150577f3bf8719f45399c4af2d88c94b
Contract's current owner address	0xcaf28025150577f3bf8719f45399c4af2d88c94b

SHELBY Token Token Distribution

The top 100 holders collectively own 86.51% (865,130,678.72 Tokens) of SHELBY Token

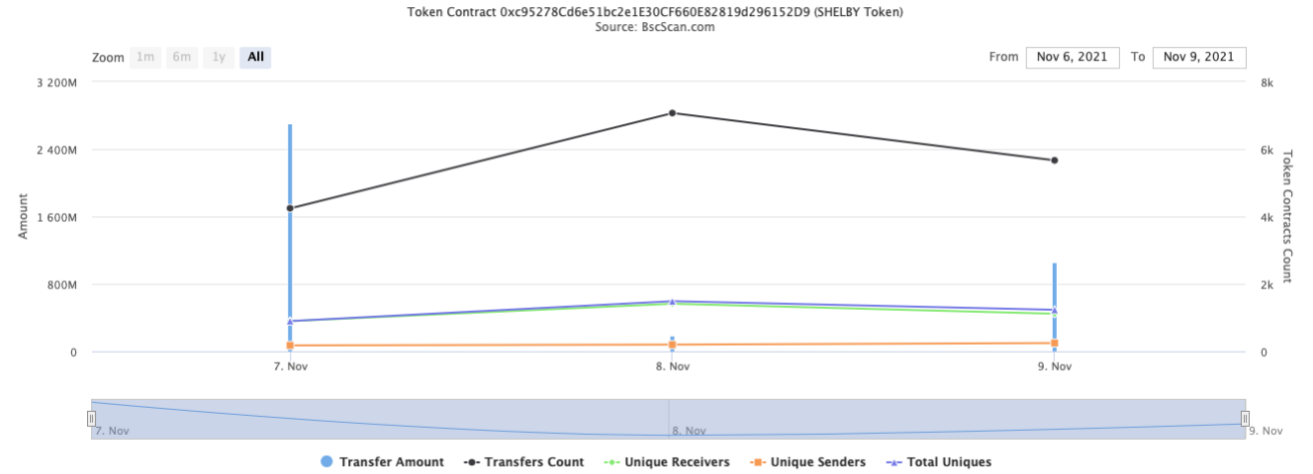
Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 2,212

(A total of 865,130,678.72 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

SHELBY Token Contract Interaction Details

Time Series: Token Contract Overview

Sun 7, Nov 2021 - Tue 9, Nov 2021



SHELBY Token Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	KIPS: Locked Wallet	300,000,000	30.0000%
2	0x494ce3ede28f7d0a81010086438165bdb95f811b	150,000,000	15.0000%
3	0x43616bb32918bfe14416d77765f9beb9a04a41a1	97,137,721.999999999999999504	9.7138%
4	PancakeSwap V2: SBY 5	47,049,655.828987798926020629	4.7050%
5	0x609814a28ffb6322cccee7faff3a010212724396	20,000,000	2.0000%
6	0x38062097a720afacc6304446d13a26f5fc361e67	13,280,152.183224877272038775	1.3280%
7	0x3214f015a1348491fe2fd34a0462d58160405610	10,000,000	1.0000%
8	0xe8a2190d41da9ccef12ffc4396a1938b18d9e395	10,000,000	1.0000%
9	0x99850ccab63ca1cfe09d61b3a5f8a1a1de98e959	10,000,000	1.0000%
10	0xc4cf5495293a22e5be5c2da6ec12c00eeecd564a	10,000,000	1.0000%

Contract functions details

+ [Int] IUniswapV2Router01
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)
  - [Ext] removeLiquidity #
  - [Ext] removeLiquidityETH #
  - [Ext] removeLiquidityWithPermit #
  - [Ext] removeLiquidityETHWithPermit #
  - [Ext] swapExactTokensForTokens #
  - [Ext] swapTokensForExactTokens #
  - [Ext] swapExactETHForTokens ($)
  - [Ext] swapTokensForExactETH #
  - [Ext] swapExactTokensForETH #
  - [Ext] swapETHForExactTokens ($)
  - [Ext] quote
  - [Ext] getAmountOut
  - [Ext] getAmountIn
  - [Ext] getAmountsOut
  - [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
  - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
  - [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
  - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ [Int] IUniswapV2Factory
  - [Ext] feeTo
  - [Ext] feeToSetter
  - [Ext] getPair
  - [Ext] allPairs
  - [Ext] allPairsLength
  - [Ext] createPair #
  - [Ext] setFeeTo #
  - [Ext] setFeeToSetter #

+ [Int] IUniswapV2Pair
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transfer #
  - [Ext] transferFrom #
  - [Ext] DOMAIN_SEPARATOR
  - [Ext] PERMIT_TYPEHASH
  - [Ext] nonces

- [Ext] permit #
- [Ext] MINIMUM_LIQUIDITY
- [Ext] factory
- [Ext] token0
- [Ext] token1
- [Ext] getReserves
- [Ext] price0CumulativeLast
- [Ext] price1CumulativeLast
- [Ext] kLast
- [Ext] mint #
- [Ext] burn #
- [Ext] swap #
- [Ext] skim #
- [Ext] sync #
- [Ext] initialize #

+ [Lib] IterableMapping
- [Int] get
- [Int] getIndexOfKey
- [Int] getKeyAtIndex
- [Int] size
- [Int] set #
- [Int] remove #

+ [Int] DividendPayingTokenOptionalInterface
- [Ext] withdrawableDividendOf
- [Ext] withdrawnDividendOf
- [Ext] accumulativeDividendOf

+ [Int] DividendPayingTokenInterface
- [Ext] dividendOf
- [Ext] distributeDividends ($)
- [Ext] withdrawDividend #

+ [Lib] SafeMathInt
- [Int] mul
- [Int] div
- [Int] sub
- [Int] add
- [Int] abs
- [Int] toUint256Safe

+ [Lib] SafeMathUint
- [Int] toInt256Safe

+ [Lib] SafeMath
- [Int] tryAdd
- [Int] trySub
- [Int] tryMul
- [Int] tryDiv
- [Int] tryMod
- [Int] add
- [Int] sub
- [Int] mul
- [Int] div

  - [Int] mod
  - [Int] sub
  - [Int] div
  - [Int] mod

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
  - [Prv] _setOwner #

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Int] IERC20Metadata (IERC20)
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals

+ ERC20 (Context, IERC20, IERC20Metadata)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _approve #
  - [Int] _beforeTokenTransfer #
  - [Int] _afterTokenTransfer #

+ DividendPayingToken (ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface)
  - [Pub] <Constructor> #
    - modifiers: ERC20

- [Ext] <Fallback> ($)
- [Pub] distributeDividends ($)
- [Pub] withdrawDividend #
- [Int] _withdrawDividendOfUser #
- [Pub] dividendOf
- [Pub] withdrawableDividendOf
- [Pub] withdrawnDividendOf
- [Pub] accumulativeDividendOf
- [Int] _transfer #
- [Int] _mint #
- [Int] _burn #
- [Int] _setBalance #

+ TOKEN (ERC20, Ownable)
- [Pub] <Constructor> #
  - modifiers: ERC20
- [Ext] <Fallback> ($)
- [Pub] updateDividendTracker #
  - modifiers: onlyOwner
- [Pub] updateUniswapV2Router #
  - modifiers: onlyOwner
- [Pub] excludeFromFees #
  - modifiers: onlyOwner
- [Pub] excludeMultipleAccountsFromFees #
  - modifiers: onlyOwner
- [Ext] setMarketingWallet #
  - modifiers: onlyOwner
- [Ext] setRewardsFee #
  - modifiers: onlyOwner
- [Ext] setLiquidityFee #
  - modifiers: onlyOwner
- [Ext] setMarketingFee #
  - modifiers: onlyOwner
- [Pub] setAutomatedMarketMakerPair #
  - modifiers: onlyOwner
- [Ext] blacklistAddress #
  - modifiers: onlyOwner
- [Prv] _setAutomatedMarketMakerPair #
- [Pub] updateGasForProcessing #
  - modifiers: onlyOwner
- [Ext] updateClaimWait #
  - modifiers: onlyOwner
- [Ext] getClaimWait
- [Ext] getTotalDividendsDistributed
- [Pub] isExcludedFromFees
- [Pub] withdrawableDividendOf
- [Pub] dividendTokenBalanceOf
- [Ext] excludeFromDividends #
  - modifiers: onlyOwner
- [Ext] getAccountDividendsInfo
- [Ext] getAccountDividendsInfoAtIndex
- [Ext] processDividendTracker #
- [Ext] claim #
- [Ext] getLastProcessedIndex
- [Ext] getNumberOfDividendTokenHolders

  - [Int] _transfer #
  - [Prv] swapAndLiquify #
  - [Prv] swapTokensForEth #
  - [Prv] addLiquidity #
  - [Prv] swapAndSendDividends #

+ TOKENDividendTracker (Ownable, DividendPayingToken)
  - [Pub] <Constructor> #
    - modifiers: DividendPayingToken
  - [Int] _transfer
  - [Pub] withdrawDividend
  - [Ext] excludeFromDividends #
    - modifiers: onlyOwner
  - [Ext] updateClaimWait #
    - modifiers: onlyOwner
  - [Ext] getLastProcessedIndex
  - [Ext] getNumberOfTokenHolders
  - [Pub] getAccount
  - [Pub] getAccountAtIndex
  - [Prv] canAutoClaim
  - [Ext] setBalance #
    - modifiers: onlyOwner
  - [Pub] process #
  - [Pub] processAccount #
    - modifiers: onlyOwner

($) = payable function
= non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issues
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user 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.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- The function `excludeMultipleAccountsFromFees()` uses the loop to exclude multiple accounts from fees. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

Recommendation:

Be careful about accounts array length.

Notes:

- Owner can change dividend tracker to not audited and some functions may work in different ways.

Owner privileges (In the period when the owner is not renounced)

- Owner can change dividend tracker.
- Owner can change Uniswap router address.
- Owner can exclude from the fees.
- Owner can exclude and include addresses in automatedMarketMakerPairs array.
- Owner can change rewards, marketing, liquidity fee.
- Owner can blacklist addresses.
- Owner can open trading.
- Owner can change gas for processing.
- Owner can update claimWait value.
- Owner can change marketing wallet.
- Owner can exclude from dividends.

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope. The further transfers and operations with the funds raise are not related to this particular contract.

Liquidity locking details provided by the team:
https://mudra.website/?certificate=yes&type=0&lp=0xa7cb9729c0152c8107f8821045b84b67c9c647ea

TechRate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.