HackSafe

Smart Contract Security Audit Report

LEMON SWAP

April 2023

Security Status

★ ★ ★ ⯪ ☆

www.hacksafe.io

Audit Details

Audited project
LEMON SWAP

Deployer address
0x161a61bc66f625277da70a38bb7307c89fc92836

Client contacts
LEMON SWAP team

Blockchain
Binance smart chain

Website
Not Provided

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) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe 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 HackSafe 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, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, 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.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by LEMON SWAP to perform an audit of smart contracts:

- https://bscscan.com/token/0x86A611fa791C22f91f38E49dEa494A85ae2dCbc0#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 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.

Contract Details

Token contract details for 07.04.2023

Type	: MEME
Contract name	: LemonSwap
Contract address	: 0x86A611fa791C22f91f38E49dEa494A85ae2dCbc0
Total supply	: 1,000,000,000,000
Token Ticker	: Lemon
Decimals	: 18
Token Holders	: 69
Transactions count	: 3,469
Compiler version	: v0.8.7+commit.e28d00a7
Contract deployer address	: 0x161a61bc66f625277da70a38bb7307c89fc92836
Owner address	: 0X6186b2b3dc4600a1281d58d377a386527a01e81b

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are **“Secure”.** This token contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.

Insecure	Poor secured	Secure	Well-secured

You are here 

We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 0 critical, 0 high, 2 medium and 0 low and some very low-level issues. These issues are not critical ones.

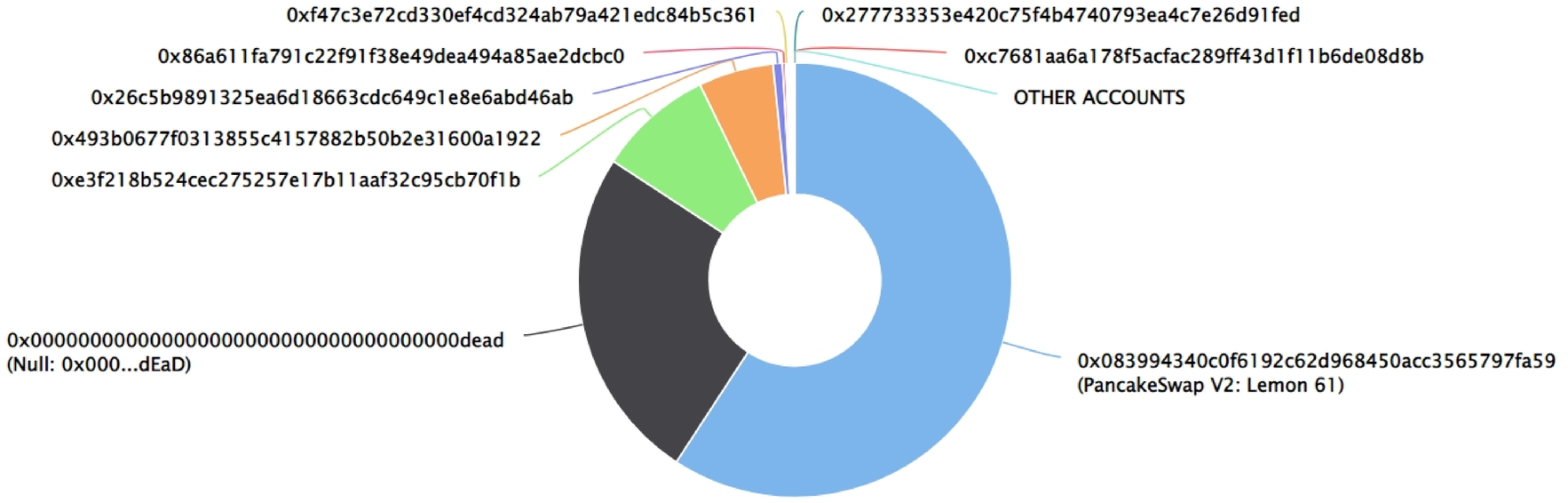
LEMON SWAP Token Distribution

The top 100 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of Lemon Swap

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 69

Lemon Swap Top 100 Token Holders

Source: BscScan.com



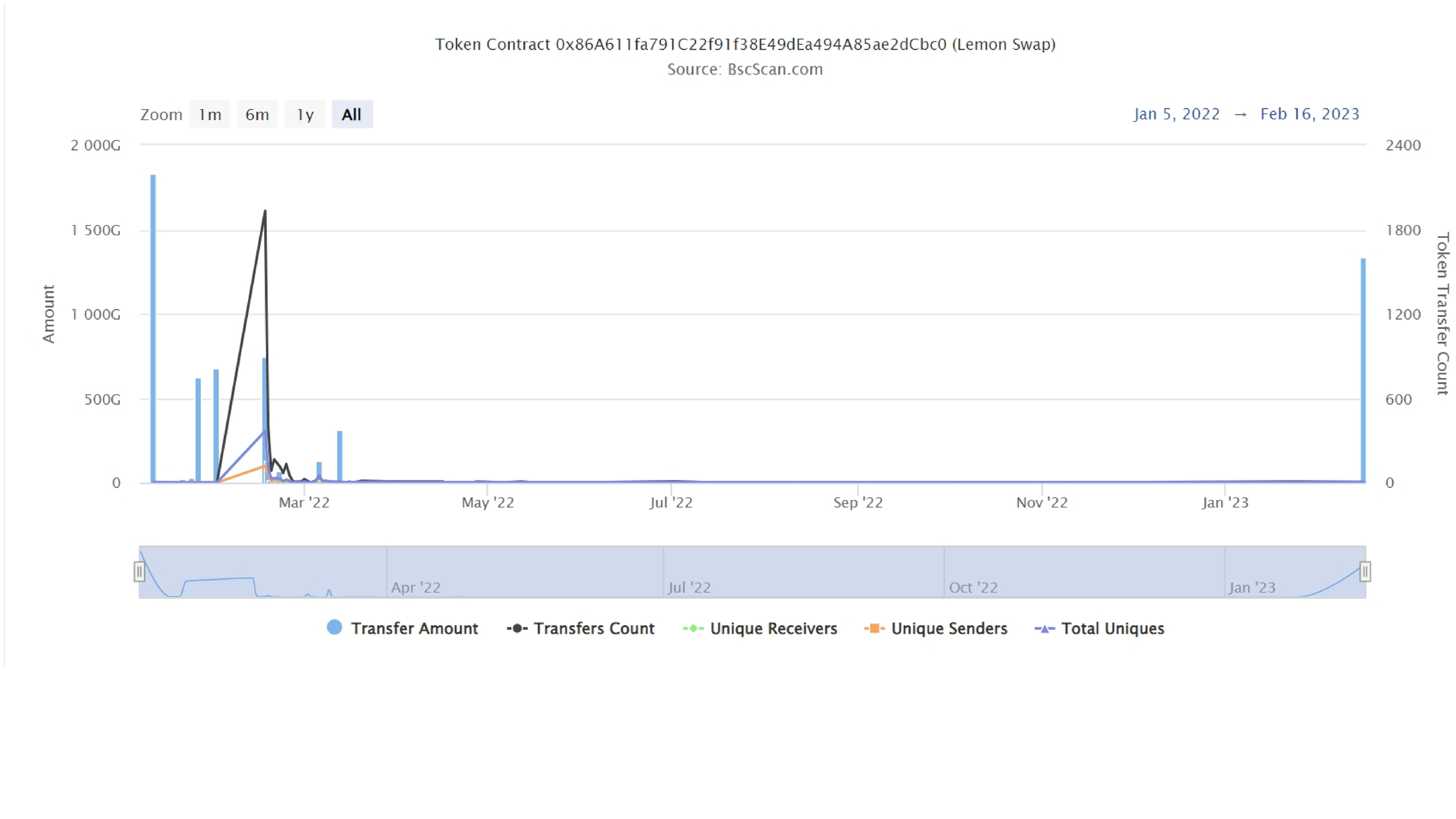
LEMON SWAP Token Top 20 Token Holders

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

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: Lemon 61	592,159,762,019.552899404586258029	59.2160%
2	Null: 0x000...dEaD	250,000,000,000	25.0000%
3	0xe3f218b524cec275257e17b11aaf32c95cb70f1b	85,892,343,404.7632639532	8.5892%
4	0x493b0677f0313855c4157882b50b2e31600a1922	55,769,269,295.551418776843537116	5.5769%
5	0x26c5b9891325ea6d18663cdc649c1e8e6abd46ab	7,000,000,000	0.7000%
6	0x86a611fa791c22f91f38e49dea494a85ae2dcbc0	2,339,340,457.417960819811975276	0.2339%
7	0xf47c3e72cd330ef4cd324ab79a421edc84b5c361	1,307,759,410	0.1308%
8	0x0ed943ce24baebf257488771759f9bf482c39706	577,303,867.419383635493671735	0.0577%
9	0xb2ca43a004fbb60e6d236e12d85fa475401031d3	540,000,000	0.0540%
10	0xcdc7f867ebfac7f95fb5c23669fcc6931ec332c3	300,000,400	0.0300%
11	0xad473621c96d7486a45971ed02e52b0b92aec3fb	280,000,000	0.0280%
12	0x714c332c906629aa4e84e6b25ee1e0ffec5f0013	280,000,000	0.0280%
13	0x2dfc0d4b37cf78b021594f43d6972116e014a77a	269,152,255.869630050353181692	0.0269%
14	0x03d3b4e706fed4a7cf4edd54fefeaef9f0c4fda1	258,662,231.864754690672427104	0.0259%
15	0xdc29ab1772304cf9068bb261c9fd29180330f63f	217,862,821.687024709630273658	0.0218%
16	0x8d0dcf108b4eb8eb5237567818657d6542788701	210,000,000	0.0210%
17	0xf9e39a92cc68b8a432b07eecfca740d679a7facb	175,443,465.485262904829951864	0.0175%
18	0x90db20503a72e9db727b1db44821c706c8b41775	165,045,565.434050348137979985	0.0165%
19	0xcf28a4dfb7a13634acea85fe313448907ee1e94d	140,000,000	0.0140%
20	0xb7bec6025afb426b79ea6d2716e7ffd0d7d2fc30	140,000,000	0.0140%

LEMON SWAP Token Distribution

LEMON SWAP Token Contract Overview



Contract functions details

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

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

+[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 #

Contract functions details

-[Int] _mint #
-[Int] _burn #
-[Int] _approve #
-[Int] _beforeTokenTransfer #
-[Int] _afterTokenTransfer #

+[Lib] SafeMath
    -[Int] add
    -[Int] sub
    -[Int] sub
    -[Int] mul
    -[Int] div
    -[Int] div
    -[Int] mod
    -[Int] mod

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

+[Lib] SafeMathUint
    -[Int] toInt256Safe

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

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

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

Contract functions details

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

+[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

Contract functions details

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

+[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 ($)

Contract functions details

-[Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

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

+ LemonSwapDividendTracker (DividendPayingToken, Ownable)
-[Pub] <Constructor> #
- modifiers: DividendPayingToken
-[Pub] decimals
-[Int] _transfer
-[Pub] getAllowCustomTokens
-[Ext] setAllowCustomTokens #
- modifiers: onlyOwner
-[Ext] excludeFromDividends #
- modifiers: onlyOwner
-[Pub] isExcludedFromDividends
-[Ext] updateClaimWait #
- modifiers: onlyOwner
-[Ext] updateMinimumTokenBalanceForDividends #
- modifiers: onlyOwner
-[Ext] getLastProcessedIndex
-[Ext] getNumberOfTokenHolders
-[Pub] getAccount
-[Pub] getAccountAtIndex
-[Pvt] canAutoClaim
-[Ext] setBalance #
- modifiers: onlyOwner
-[Pub] process #
-[Pub] processAccount #
- modifiers: onlyOwner
-[Pub] updateUniswapV2Router #
- modifiers: onlyOwner
-[Pub] updatePayoutToken #

Contract functions details

  - modifiers: onlyOwner
-[Pub] updatePayoutToken #
  - modifiers: onlyOwner
-[Pub] getPayoutToken
-[Pub] updateAllowTokens #
  - modifiers: onlyOwner
-[Pub] getAllowTokens
-[Int] _withdrawDividendOfUser #

+ LemonSwap (ERC20, Ownable)
-[Pub] <Constructor> #
  - modifiers: ERC20
-[Pub] decimals
-[Pub] setMaxWalletRate #
  - modifiers: onlyOwner
-[Ext] <Fallback> ($)
-[Ext] setSwapTokensAtAmount #
  - modifiers: onlyOwner
-[Pub] updateDividendTracker #
  - modifiers: onlyOwner
-[Pub] updateUniswapV2Router #
  - modifiers: onlyOwner
-[Pub] excludeFromFees #
  - modifiers: onlyOwner
-[Pub] excludeMultipleAccountsFromFees #
  - modifiers: onlyOwner
-[Ext] updateMarketingWallet #
  - modifiers: onlyOwner
-[Ext] updateDevWallet #
  - modifiers: onlyOwner
-[Int] isFeeAcceptable
-[Ext] setMarketingSellFee #
  - modifiers: onlyOwner
-[Ext] setMarketingBuyFee #
  - modifiers: onlyOwner
-[Ext] setDevSellFee #
  - modifiers: onlyOwner
-[Ext] setDevBuyFee #

Contract functions details

- modifiers: onlyOwner
-[Ext] setLiquiditySellFee #
 modifiers: onlyOwner
-[Ext] setLiquidityBuyFee #
 - modifiers: onlyOwner
-[Ext] setReflectionSellFee #
 - modifiers: onlyOwner
-[Ext] setReflectionBuyFee #
 - modifiers: onlyOwner
-[Pub] setAutomatedMarketMakerPair #
 - modifiers: onlyOwner
-[Pvt] _setAutomatedMarketMakerPair #
-[Pub] updateGasForProcessing #
 - modifiers: onlyOwner
-[Ext] updateClaimWait #
 - modifiers: onlyOwner
-[Ext] getClaimWait
-[Ext] updateMinimumTokenBalanceForDividends #
 - modifiers: onlyOwner
-[Ext] getMinimumTokenBalanceForDividends
-[Ext] getTotalDividendsDistributed
-[Pub] isExcludedFromFees
-[Pub] withdrawableDividendOf
-[Pub] dividendTokenBalanceOf
-[Ext] excludeFromDividends #
 - modifiers: onlyOwner
-[Pub] isExcludedFromDividends
-[Ext] getAccountDividendsInfo
-[Ext] getAccountDividendsInfoAtIndex
-[Ext] processDividendTracker #
-[Ext] claim #
-[Ext] claimFor #
-[Ext] getLastProcessedIndex
-[Ext] getNumberOfDividendTokenHolders
-[Int] _transfer #
 - modifiers: antiWhale
-[Pvt] swapAndLiquify #
-[Pvt] swapAndSendDividendsMarketingDev #

Contract functions details

-[Pvt] swapTokensForEth #
-[Pvt] addLiquidity #
-[Pub] setAntiBotSystemEnable #
- modifiers: onlyOwner
-[Pub] setBotSettingTime #
- modifiers: onlyOwner
-[Pub] setBotFeeMultiplicator #
- modifiers: onlyOwner
-[Pub] excludeAntibot #
- modifiers: onlyOwner
-[Pub] isBot
-[Pub] setEnableAntiwhale #
- modifiers: onlyOwner
-[Pub] maxTransferAmount
-[Pub] setMaxTransferAmountRate #
- modifiers: onlyOwner
-[Pub] updatePayoutToken #
-[Pub] getPayoutToken
-[Pub] updateAllowTokens #
- modifiers: onlyOwner
-[Pub] getAllowTokens
-[Pub] enableSwapAndLiquify #
- modifiers: onlyOwner
-[Pub] setSwapTokensAmountMax #
- modifiers: onlyOwner
-[Ext] getNativeBalance
-[Ext] getCountOfFeesToSwap
-[Ext] transferERC20Token #
- modifiers: onlyOwner
-[Pub] setExcludeAntiwhale #
- modifiers: onlyOwner
-[Pub] setExcludeMaxWallet #
- modifiers: onlyOwner

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Medium issue
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Medium issue
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

- **Critical Severity Issues**
  No critical severity issue found.
- **High Severity Issues**
  No high severity issue found.
- **Medium Severity Issues**
  Two Medium severity issues found.

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.

2. swapAndLiquify issue

- **Issue:**
  • The function swapAndLiquify() do not returns after emitting zero event on “(tokens > balanceOf(address(this)))”.
- **Recommendation**
  Be careful about accounts array length.

- **Low Severity Issues**
  No low severity issue found.

Centralization

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

- LEMON SWAP Contract:
    - Owner can change _maxWalletSize.
    - Owner can change swapTokensAtAmount.
    - Owner can change dividend tracker.
    - Owner can change Uniswap router address.
    - Owner can exclude from the fees.
    - Owner can change marketing and dev wallet addresses.
    - Owner can enable/disable antiWahle.
    - Owner can change maxTransferAmountRate.
    - Owner can change allow tokens.
    - Owner can enable/disable swap and liquify.
    - Owner can change swapTokensAtAmountMax.
    - Owner can withdraw contract ERC20 tokens. • Owner can exclude from antiWahle and maxWallet.
    - Owner can change fees.
    - Owner can exclude and include addresses in automatedMarketMakerPairs array.
    - Owner can change gas for processing.
    - Owner can change claim wait value.
    - Owner can change minimum tokens for dividends.
    - Owner can exclude addresses from dividends.
    - Owner can enable/disable antibotSystemEnable.
    - Owner can change bot settings time.
    - Owner can change _botIncreaseFee.
    - Owner can exclude from antiBot.

Conclusion

Smart contract contains medium severity issues! Liquidity pair contract's security is not checked due to out of scope.

HackSafe 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.