HackSafe

Smart Contract Security Audit Report

Yarloo

March 2023

Security Status

★★★★☆

www.hacksafe.io

Audit Details

Audited project
Yarloo

Deployer address
0x719508587ca181e2ea738e48f6fb83995b7cd77e

Client contacts
Yarloo

Blockchain
Binance Smart Chain

Website
https://www.yarloo.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) (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 Yarlooto perform an audit of smart contracts:

- https://bscscan.com/address/0x843cbc1732ae7d7ba0533c6380989dacec315ffe#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 30.03.2023

Type	: Gaming
Contract name	: Yarloo
Contract address	: 0x843CbC1732aE7D7ba0533C6380989DACec315FfE
Total supply	: 25,000,000
Token Ticker	: YARL
Decimals	: 18
Token Holders	: 4,541
Top 100 token holder's dominance	: 96.42%
Transactions count	: 50,642
Compiler version	: v0.8.9+commit.e5eed63a
Contract deployer address	: 0x719508587ca181e2ea738e48f6fb83995b7cd77e
Owner address	: 0x719508587ca181e2ea738e48f6fb83995b7cd77e

Social profiles

Twitter Profile	: https://twitter.com/yarloo_rst
Telegram Profile	: https://t.me/yarloochat
Reddit Profile	: https://www.reddit.com/r/yarloo/
Medium Profile	: https://medium.com/@yarloo
LinkedIn Profile	: https://www.linkedin.com/company/yarloo
Coinmarketcap Profile	: https://coinmarketcap.com/currencies/yarloo/
Coingecko Profile	: https://www.coingecko.com/en/coins/yarloo

Claimed Smart Contract Features

Claimed Feature Detail	Our Observation
Tokenomics : • **Name** : ParmaFanToken • **Symbol** : YARL • **Decimals** : 18 • **Protocol** : Gaming • **Total supply** : 25,000,000 • **Contract address** 0x843CbC1732aE7D7ba0533C6380989DACec315FfE	YES, this is valid.

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, 1 medium and low and some very low-level issues. These issues are not critical ones.

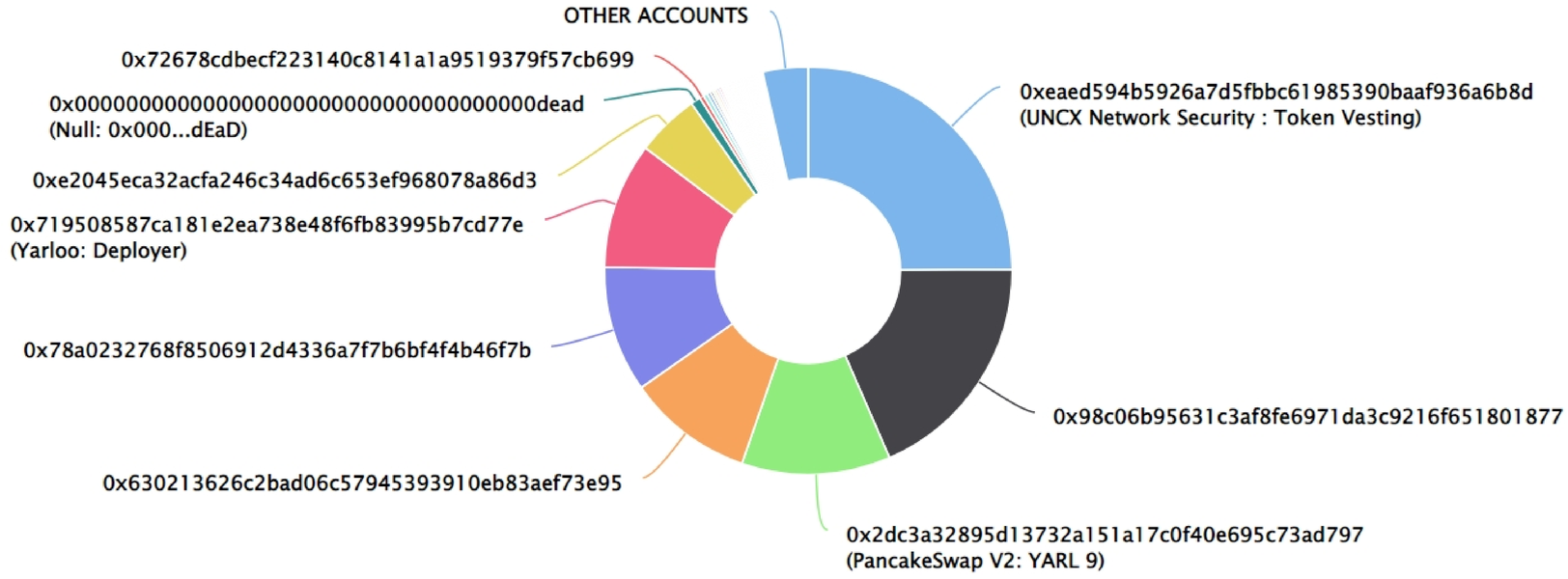
Yarloo Token Distribution

The top 100 holders collectively own 96.42% (24,103,830.25 Tokens) of Yarloo

Token Total Supply: 25,000,000.00 Token | Total Token Holders: 4,541

Yarloo Top 100 Token Holders

Source: BscScan.com



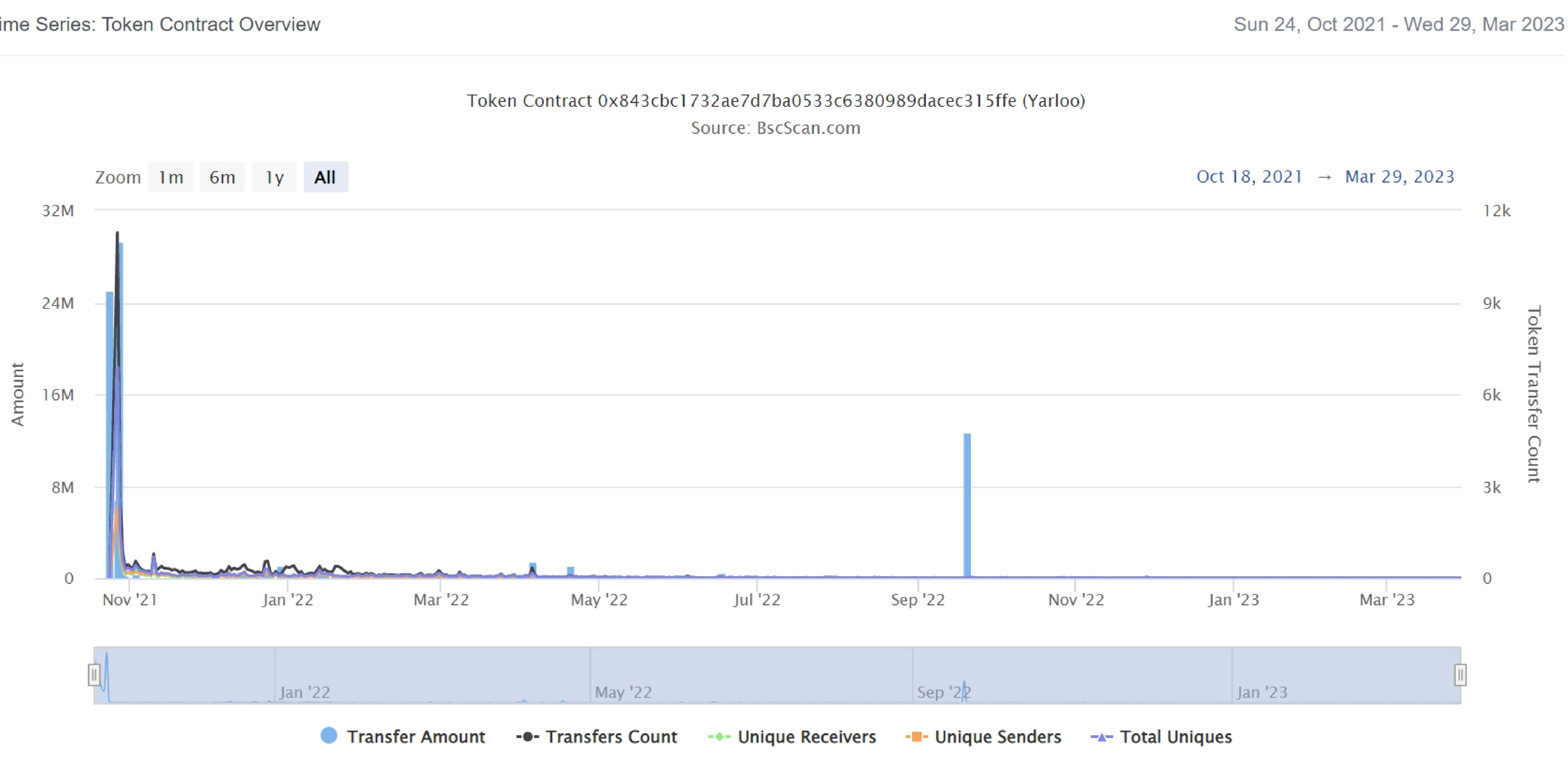
Yarloo Token Top 20 Token Holders

(A total of 24,103,830.25 tokens held by the top 100 accounts from the total supply of 25,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	UNCX Network Security : Token Vesting	6,224,375	24.8975%
2	0x98c06b95631c3af8fe6971da3c9216f651801877	4,643,848.792755415017578634	18.5754%
3	PancakeSwap V2: YARL 9	2,955,254.631441706194212906	11.8210%
4	0x630213626c2bad06c57945393910eb83aef73e95	2,500,000	10.0000%
5	0x78a0232768f8506912d4336a7f7b6bf4f4b46f7b	2,500,000	10.0000%
6	Yarloo: Deployer	2,499,979.229396389025841111	9.9999%
7	0xe2045eca32acfa246c34ad6c653ef968078a86d3	1,250,000	5.0000%
8	Null: 0x000...dEaD	195,000	0.7800%
9	0x72678cdbecf223140c8141a1a9519379f57cb699	86,433.192789999999996	0.3457%
10	0x619f07316e352378b48ca8934224e536ae752297	78,019.138019999999435	0.3121%
11	0x07b475b45659d941fbd81a1b194a25322ce7c949	67,062.36528204832551134	0.2682%
12	0x919564124877a1cd21f5993ac11402e52510ed32	53,484.236766237026614523	0.2139%
13	0xa541e31a2aa1d9717b64786d3c7b2406af89c952	47,712.441330625792481467	0.1908%
14	Bitmart: Wallet 2	47,431.095983320568610359	0.1897%
15	0x38325956905280a626d50c07a5e9642930cdfa33	45,610.853424650000949	0.1824%
16	0xb7ce32ed4d2bd47ce951c2141fe16c351d60226e	44,358.1417131851504888	0.1774%
17	0xb120cc62f97a31a77f699bd5b290499abe5c5148	40,000	0.1600%
18	0x4189aef8999c1f6a453dbda137d4716bd72ac03d	39,954.509343829879914838	0.1598%
19	0x53d70346a770899109578b7592aebf86dd09997e	39,500	0.1580%
20	0x2259080a998ac09ab443018288322e30514df5e5	31,350.442982231007625408	0.1254%

Yarloo Token Distribution

Yarloo Token Contract Overview



Contract functions details

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

+[Int] IBEP20
    -[Ext] totalSupply
    -[Ext] balanceOf
    -[Ext] transfer #
    -[Ext] allowance
    -[Ext] approve #
    -[Ext] transferFrom #

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

+[Lib] Address
    -[Int] isContract
    -[Int] sendValue #
    -[Int] functionCall #
    -[Int] functionCall #
    -[Int] functionCallWithValue #
    -[Int] functionCallWithValue #
    -[Pvt] _functionCallWithValue #

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

Contract functions details

-[Pub] lock #
  - modifiers: onlyOwner
-[Pub] unlock #

+[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] burn #
-[Ext] swap #
-[Ext] skim #

Contract functions details

-[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 ($)
-[Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+TransactionThrottler (Ownable)
-[Pub] <Constructor> #
-[Ext] setTradingStart #
- modifiers: onlyOwner
-[Ext] setMaxTransferAmount #
- modifiers: onlyOwner
-[Ext] setRestrictionActive #
- modifiers: onlyOwner

Contract functions details

-[Ext] unthrottleAccount #
  - modifiers: onlyOwner
-[Ext] isUnthrottled
-[Ext] whitelistAccount #
  - modifiers: onlyOwner
-[Ext] isWhitelisted

+Yarloo (Context, IBEP20, Ownable, TransactionThrottler)
-[Pub] <Constructor> #
-[Pub] name
-[Pub] symbol
-[Pub] decimals
-[Pub] totalSupply
-[Pub] balanceOf
-[Pub] transfer #
-[Pub] allowance
-[Pub] approve #
-[Pub] transferFrom #
-[Pub] increaseAllowance #
-[Pub] decreaseAllowance #
-[Pub] isExcludedFromReward
-[Pub] totalFees
-[Pub] totalBurn
-[Pub] totalDonationBNB
-[Pub] minimumTokensBeforeSwapAmount
-[Pub] deliver #
-[Pub] reflectionFromToken
-[Pub] tokenFromReflection
-[Pub] excludeFromReward #
  - modifiers: onlyOwner
-[Ext] includeInReward #
  - modifiers: onlyOwner
-[Pvt] _approve #
-[Pvt] _transfer #
  - modifiers: transactionThrottler
-[Pvt] swapAndLiquify #
  - modifiers: lockTheSwap
-[Pvt] swapTokensForBNB #

Contract functions details

-[Pvt] _tokenTransfer #
-[Pvt] _transferStandard #
-[Pvt] _transferToExcluded #
-[Pvt] _transferFromExcluded #
-[Pvt] _transferBothExcluded #
-[Pvt] _reflectFee #
-[Pvt] _getValues
-[Pvt] _getTValues
-[Pvt] _getRValues
-[Pvt] _getRate
-[Pvt] _getCurrentSupply
-[Pvt] _takeLiquidity #
-[Pvt] calculateTaxFee
-[Pvt] calculateBurnFee
-[Pvt] calculateLiquidityFee
-[Pvt] removeAllFee #
-[Pvt] restoreAllFee #
-[Pub] isExcludedFromFee
-[Pub] excludeFromFee #
  - modifiers: onlyOwner
-[Pub] includeInFee #
  - modifiers: onlyOwner
-[Ext] setMarketingFeePercent #
  - modifiers: onlyOwner
-[Ext] setMarketingAddFee #
  - modifiers: onlyOwner
-[Ext] setMarketingAdd #
  - modifiers: onlyOwner
-[Ext] setDeployerAdd #
  - modifiers: onlyOwner
-[Ext] setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
-[Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
-[Pvt] TransferMarketingBNB #
-[Ext] <Fallback> ($)

Contract functions details

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Compiler error	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 Issues
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.	Passed
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**
  One Medium severity issues found.

1. Out of Gas

- **Issue:**
  - The function includeInReward() uses the loop to find and remove addresses from the _excluded list. Function will be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

  - The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

- **Recommendation**
  - Check that the excluded array length is not too big.

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

  Notes: addLiquidity function is not used.

Centralization

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

- Yarloo Contract:
    - Owner can start trading.
    - Owner can change the maximum transaction amount.
    - Owner can enable/disable _restrictionActive.
    - Owner can unthrottle and whitelist accounts.
    - Owner can exclude from the fee.
    - Owner can change marketing and marketingAdd fee.
    - Owner can change marketing and deployer address.
    - Owner can change minimum number of tokens to add to liquidity.
    - Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.

Conclusion

Smart contract contains medium severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

Liquidity locking details NOT provided by the team.

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.