SOLIDITY FINANCE

MYSTERY

Smart Contract Audit Report

AUDIT SUMMARY

MYSTERY is a new BEP-20 token that is an automatic liquidity providing protocol.

For this audit, we reviewed the Mystery contract provided to us by the project team.

AUDIT FINDINGS

> *Please ensure trust in the team prior to investing as they have substantial control in the ecosystem.*
> *The team has successfully completed KYC procedures with KYC capital.*
> *Date: March 2nd, 2022.*

CONTRACT OVERVIEW

- *The total supply of the token is set to 1 billion $MYST [1,000,000,000].*
- *No mint or burn functions exist, though the circulating supply can be decreased by sending tokens to the 0x..dead address.*
- *There was no token allocation for our team to analyze as the contract has yet to be deployed to the mainnet.*

- *The owner must manually enable trading in order for all trading to take place on the*

enabled. Once trading has been enabled, it can never be disabled.

- *Users who are attempting to buy tokens within the same block that trading was enabled will be taxed at a 99.9% rate.*
- *The contract enforces a maximum transaction amount (determined by the owner) which imposes a limit to the number of tokens that can be transferred during any given transaction.*
- *The contract enforces a maximum wallet amount that prevents a transfer from occurring when the following conditions are met:*
  - *The recipient is not the Pair address or the 0x..dead address.*
  - *The recipient is not excluded from the limit.*
  - *The recipient's token balance will exceed the limit number of tokens (determined by the owner) after the transfer occurs.*
- *The contract enforces an antidump mechanism that increases total fees by a percentage (determined by the owner) when selling tokens if the price impact that the number of tokens a user has sold (within the last 30 minutes) will have on the liquidity pool is more than the antidump threshold value (also determined by the owner).*
- *The contract includes functionality to interact with an external antisniper contract. This antisniper contract was out of scope for the purpose of this audit.*
- *There is a Liquidity fee, Marketing fee, and Dev fee on all transfers where neither the sender nor the recipient is excluded from fees.*
- *Fees will not be charged on transfers while the contract is currently performing an automatic liquidity add.*
- *Total fees are increased by a value (determined by the owner) when selling tokens to Pancakeswap (or any other approved DEX). Total fees are decreased by this same value on all other transactions.*
- *If a user is attempting to sell tokens after having bought tokens for the first time within 24 hours, total fees will be increased by a percentage (determined by the owner).*
- *The Liquidity fee charged during transfers is stored in the contract address. The tokens are swapped for BNB for the purpose of funding Pancakeswap liquidity when the following conditions are met:*
  - *The automatic liquidity add functionality is enabled by the team.*
  - *The threshold number of tokens in the contract address (determined by the owner) has been reached.*
  - *The contract is not currently performing an automatic liquidity add.*
  - *The sender is not performing a buy transaction via Pancakeswap.*

  - *The sender is not excluded from fees.*
  - *The sum of the current transfer fees is greater than 0.*
- *Liquidity-adds are automatically performed by selling the tokens collected as fees, pairing the received BNB with the token, and adding it as liquidity to the BNB pair.*
- *The LP tokens received through this process are sent to the Liquidity address controlled by the team. We recommend that the team lock these newly acquired LP tokens.*
- *The tokens collected from the Dev fee and Marketing fee are swapped for BNB and are sent to the team's Dev wallet and Marketing wallet respectively.*

- *As the contract is implemented with Solidity v0.8.x, it is protected from overflows/underflows.*
- *The contract complies with the BEP-20 token standard.*

Ownership Controls:

- *The owner can modify the Liquidity fee, Marketing fee, and Dev fee to any percentages as long as the total percentage combined does not exceed 25%.*
- *The owner can exclude and include accounts from transfer fees.*
- *The owner can update the maximum transaction amount and maximum wallet amount at any time.*
- *The owner can exclude/include accounts from the maximum transaction amount and maximum wallet amount limits.*
- *The owner can enable/disable automatic liquidity adds at any time.*
- *The owner can update the threshold number of tokens needed to trigger an automatic liquidity add to any value at any time.*
- *The owner can set the maximum amount of tokens from the contract that will be used for automatic liquidity adds to any value at any time.*
- *The owner can add accounts to a whitelist that will allow them to participate in transfers before trading has been enabled.*
- *The owner can update the sell percent to any value greater than 100 as long as the percent does not exceed the antidump tax and the product of this value and total fees divided by 100 does not exceed 400.*
- *The owner can set the sell period to any value up to 7 days.*
- *The owner can update the antidump period to any value up to 1 hour.*
- *The owner can update the antidump tax to any value up to 400. If set to 400, total fees will*

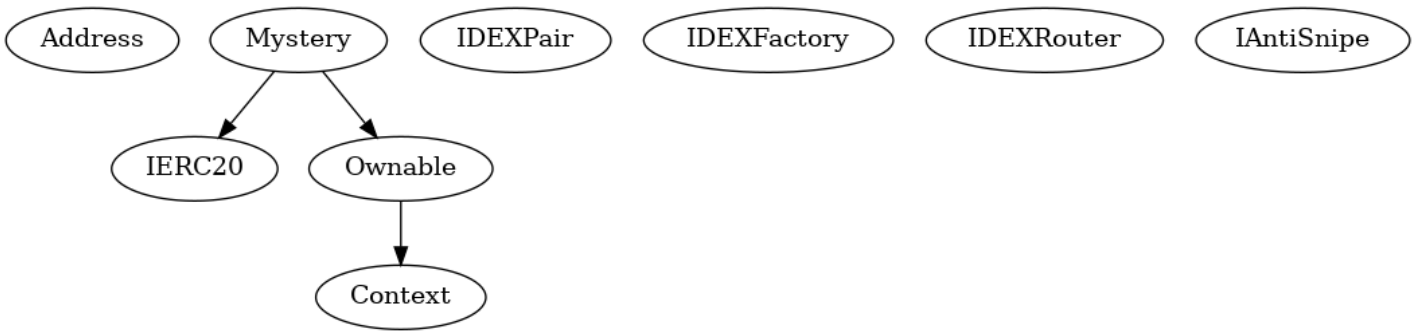
> *be increased by 4x their current value.*
> - *The owner can airdrop tokens to any addresses besides an approved DEX or the 0x00 address at any time.*
> - *The owner can enable/disable the antisniper mechanism at any time.*
> - *The owner can add any address as an approved DEX at any time.*

AUDIT RESULTS

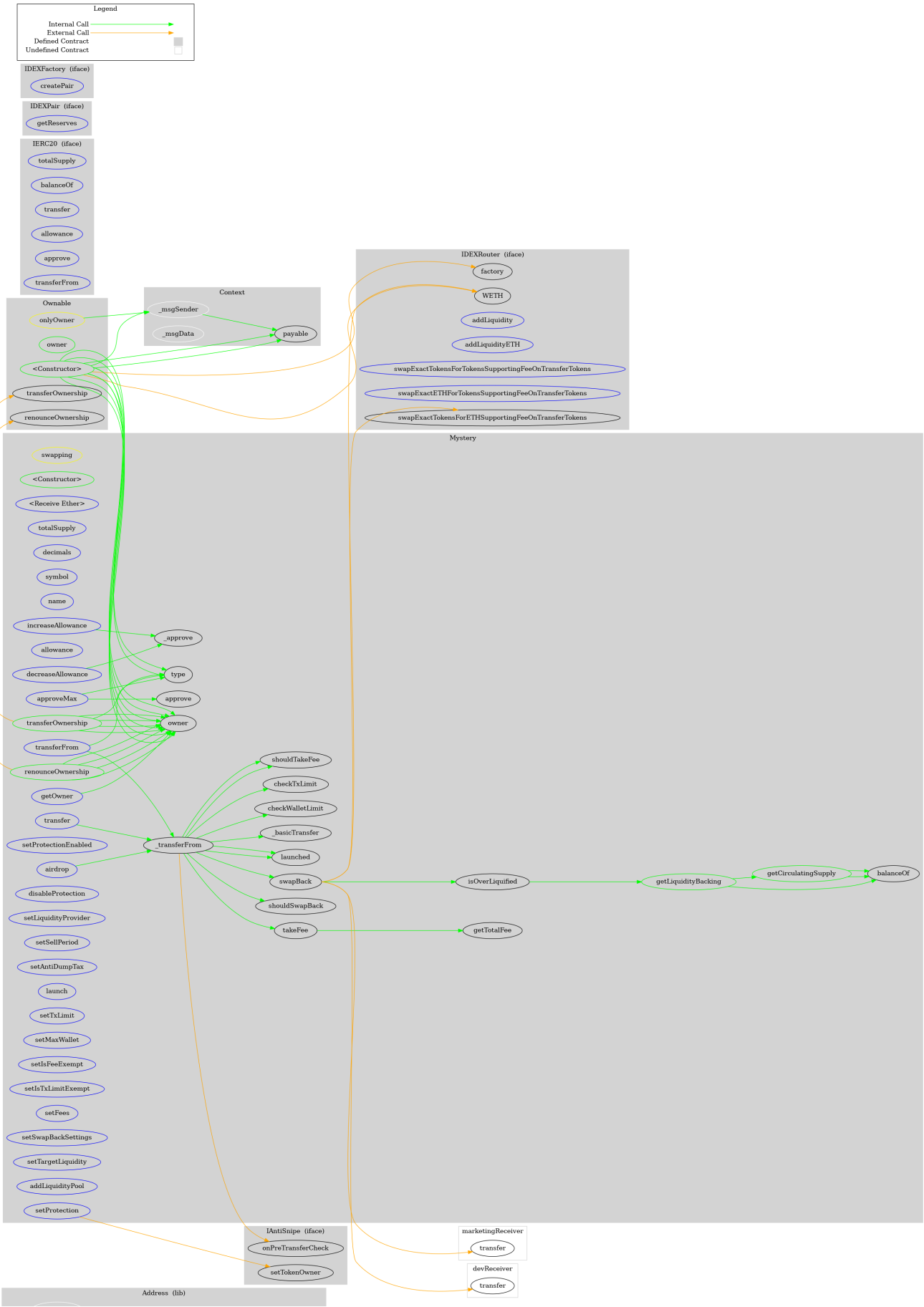
Vulnerability Category	Notes	Result
Arbitrary Jump/Storage Write	N/A	PASS
Centralization of Control	• The owner can set total fees up to 99% when selling tokens to Pancakeswap. • The recipient of the LP tokens generated through automatic liquidity adds is the Liquidity address controlled by the team.	WARNING
Compiler Issues	N/A	PASS
Delegate Call to Untrusted Contract	N/A	PASS
Dependence on Predictable Variables	N/A	PASS
Ether/Token Theft	N/A	PASS
Flash Loans	N/A	PASS
Front Running	N/A	PASS
Improper Events	N/A	PASS

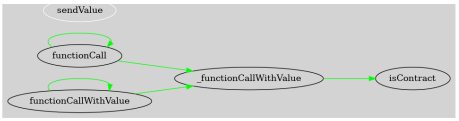
Improper Authorization Scheme	N/A	PASS
Integer Over/Underflow	N/A	PASS
Logical Issues	N/A	PASS
Oracle Issues	N/A	PASS
Outdated Compiler Version	N/A	PASS
Race Conditions	N/A	PASS
Reentrancy	N/A	PASS
Signature Issues	N/A	PASS
Unbounded Loops	N/A	PASS
Unused Code	N/A	PASS
Overall Contract Safety		PASS

INHERITANCE CHART



FUNCTION GRAPH





FUNCTIONS OVERVIEW

```
($) = payable function
# = non-constant function

Int = Internal
Ext = External
Pub = Public

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

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

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

+ [Int] IDEXPair
   - [Ext] getReserves

+ [Int] IDEXFactory
   - [Ext] createPair #

+ [Int] IDEXRouter
```

```
    - [Ext] factory
    - [Ext] WETH
    - [Ext] addLiquidity #
    - [Ext] addLiquidityETH ($)
    - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
    - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
    - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+  Ownable (Context)
    - [Pub]  #
    - [Pub] owner
    - [Pub] renounceOwnership #
       - modifiers: onlyOwner
    - [Pub] transferOwnership #
       - modifiers: onlyOwner

+ [Int] IAntiSnipe
    - [Ext] setTokenOwner #
    - [Ext] onPreTransferCheck #

+  Mystery (IERC20, Ownable)
    - [Pub]  #
    - [Ext]  ($)
    - [Ext] totalSupply
    - [Ext] decimals
    - [Ext] symbol
    - [Ext] name
    - [Ext] getOwner
    - [Pub] balanceOf
    - [Ext] allowance
    - [Pub] approve #
    - [Ext] increaseAllowance #
    - [Ext] decreaseAllowance #
    - [Int] _approve #
    - [Ext] approveMax #
    - [Ext] transfer #
    - [Ext] transferFrom #
    - [Int] _transferFrom #
    - [Int] _basicTransfer #
    - [Int] checkWalletLimit
    - [Int] checkTxLimit
    - [Int] shouldTakeFee
    - [Pub] getTotalFee
    - [Int] takeFee #
```

```
    - [Int] shouldSwapBack
    - [Int] swapBack #
      - modifiers: swapping
    - [Int] launched
    - [Pub] getCirculatingSupply
    - [Pub] getLiquidityBacking
    - [Pub] isOverLiquified
    - [Pub] transferOwnership #
      - modifiers: onlyOwner
    - [Pub] renounceOwnership #
      - modifiers: onlyOwner
    - [Ext] setProtectionEnabled #
      - modifiers: onlyOwner
    - [Ext] setProtection #
      - modifiers: onlyOwner
    - [Ext] disableProtection #
      - modifiers: onlyOwner
    - [Ext] setLiquidityProvider #
      - modifiers: onlyOwner
    - [Ext] setSellPeriod #
      - modifiers: onlyOwner
    - [Ext] setAntiDumpTax #
      - modifiers: onlyOwner
    - [Ext] launch #
      - modifiers: onlyOwner
    - [Ext] setTxLimit #
      - modifiers: onlyOwner
    - [Ext] setMaxWallet #
      - modifiers: onlyOwner
    - [Ext] setIsFeeExempt #
      - modifiers: onlyOwner
    - [Ext] setIsTxLimitExempt #
      - modifiers: onlyOwner
    - [Ext] setFees #
      - modifiers: onlyOwner
    - [Ext] setSwapBackSettings #
      - modifiers: onlyOwner
    - [Ext] setTargetLiquidity #
      - modifiers: onlyOwner
    - [Ext] addLiquidityPool #
      - modifiers: onlyOwner
    - [Ext] airdrop #
      - modifiers: onlyOwner
```

ABOUT SOLIDITY FINANCE

Solidity Finance was founded in 2020 and quickly grew to have one of the most experienced and well-equipped smart contract auditing teams in the industry. Our team has conducted 1000+ solidity smart contract audits covering all major project types and protocols, securing a total of over $10 billion U.S. dollars in on-chain value.
Our firm is well-reputed in the community and is trusted as a top smart contract auditing company for the review of solidity code, no matter how complex. Our team of experienced solidity smart contract auditors performs audits for tokens, NFTs, crowdsales, marketplaces, gambling games, financial protocols, and more!

Contact us today to get a free quote for a smart contract audit of your project!

WHAT IS A SOLIDITY AUDIT?

Typically, a smart contract audit is a comprehensive review process designed to discover logical errors, security vulnerabilities, and optimization opportunities within code. A *Solidity Audit* takes this a step further by verifying economic logic to ensure the stability of smart contracts and and highlighting privileged functionality to create a report that is easy to understand for developers and community members alike.

HOW DO I INTERPRET THE FINDINGS?

Each of our Findings will be labeled with a Severity level. We always recommend the team resolve High, Medium, and Low severity findings prior to deploying the code to the mainnet. Here is a breakdown on what each Severity level means for the project:

- **High** severity indicates that the issue puts a large number of users' funds at risk and has a high probability of exploitation, or the smart contract contains serious logical issues which can prevent the code from operating as intended.
- **Medium** severity issues are those which place at least some users' funds at risk and has a medium to high probability of exploitation.
- **Low** severity issues have a relatively minor risk association; these issues have a low probability of occuring or may have a minimal impact.
- **Informational** issues pose no immediate risk, but inform the project team of opportunities for gas optimizations and following smart contract security best practices.

GO HOME