



Smart Contract Audit

FOR
MARSU

DATED : 29 MAY 23'



MANUAL TESTING

Logical – zero swapTokensAtAmount can disable sell/transfers

Severity: **Critical**

function: updateSwapTokensAtAmount

Status: Open

Overview:

Setting swapTokensAtAmount to 0 can disable sell and transfer transactions for regular wallets (non whitelisted), this is because even if swapTokensAtAmount is set to 0, internal swap is still performed and reverts the transaction in attempt to swap 0 tokens for bnb.

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {  
    require(amount <= 4206900000000, "Cannot set swap threshold amount higher than 1% of  
tokens");  
    swapTokensAtAmount = amount * 10**_decimals;  
}
```

Suggestion

Ensure that swapTokensAtAmount is always greater than a reasonable minimum value:

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {  
    require(amount <= 4206900000000, "Cannot set swap threshold amount higher than 1% of  
tokens");  
    require(amount >= 4206900000, "Cannot set swap threshold amount higher than  
0.0001% of tokens");  
    swapTokensAtAmount = amount * 10 ** _decimals;  
}
```



AUDIT SUMMARY

Project name - MARSU

Date: 29 May, 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: Passed

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	1



USED TOOLS

Tools:

1. Manual Review: The code has undergone a line-by-line review by the Ace team.

2. ETH Test Network: All tests were conducted on the ETH Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3. Slither: The code has undergone static analysis using Slither.

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0x98ED8A59CA121DA7AB01633B963c3b8C22Ba1308>



Token Information

Name : Marsupilami Inu

Symbol : MARSU

Decimals: 9

Network: Binance smart chain

Token Type:BEP20

Token Address:

0x8FEA3bf0bcdD542EC884A72319cE420D04eE2040

Owner:

0x7320764F01B443Ed7Db15161E02650115AF5bb42
(at time of writing the audit)

Deployer:0x7320764F01B443Ed7Db15161E02650115
AF5bb42



Token Information

Fees:

Buy Fees: 8%

Sell Fees: 8%

Transfer Fees: 8%

Fees Privilege: static fees

Ownership :

0x7320764F01B443Ed7Db15161E02650115AF5bb42

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: - Initial distribution of the tokens

- excluding from fees
 - including in fees
 - enabling trades
 - changing internal swap settings
-



AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
- Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
- Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
- Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
- Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
- Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.

VULNERABILITY CHECKLIST



Return values of low-level calls



Gasless Send



Private modifier



Using block.timestamp



Multiple Sends



Re-entrancy



Using Suicide



Tautology or contradiction



Gas Limit and Loops



Timestamp Dependence



Address hardcoded



Revert/require functions



Exception Disorder



Use of tx.origin



Using inline assembly



Integer overflow/underflow



Divide before multiply



Dangerous strict equalities



Missing Zero Address Validation



Using SHA3



Compiler version not fixed



Using throw



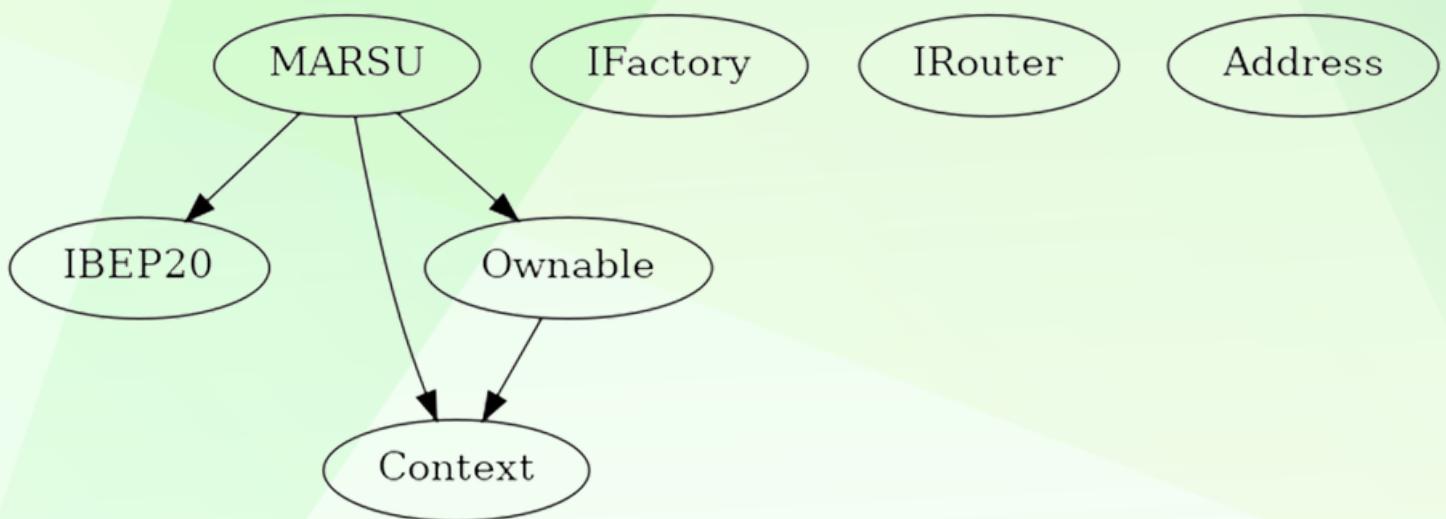
CLASSIFICATION OF RISK

Severity	Description
◆ Critical	These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
◆ High-Risk	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.
◆ Medium-Risk	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.
◆ Low-Risk	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.
◆ Gas Optimization / Suggestion	A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity	Found
◆ Critical	0
◆ High-Risk	0
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	1

INHERITANCE TREE





POINTS TO NOTE

- Fees are 0 (static)
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to limit buy/sell/transfer/wallet amounts
- Owner is not able to mint new tokens

CONTRACT ASSESSMENT

Contract	Type	Bases			
Function Name **Visibility** **Mutability** **Modifiers**					
IBEP20 Interface 					
L totalSupply External !	!	NO !			
L balanceOf External !	!	NO !			
L transfer External !	!	● NO !			
L allowance External !	!	NO !			
L approve External !	!	● NO !			
L transferFrom External !	!	● NO !			
Context Implementation 					
L _msgSender Internal 🔒					
L _msgData Internal 🔒					
Ownable Implementation Context 					
L <Constructor> Public !	●	NO !			
L owner Public !	!	NO !			
L renounceOwnership Public !	!	●	onlyOwner		
L transferOwnership Public !	!	●	onlyOwner		
L _setOwner Private 🔒	●				
IFactory Interface 					
L createPair External !	●	NO !			
IRouter Interface 					
L factory External !	!	NO !			
L WETH External !	!	NO !			
L addLiquidityETH External !	!	SW	NO !		
L swapExactTokensForETHSupportingFeeOnTransferTokens External !	●	NO !			
Address Library 					
L sendValue Internal 🔒	●				
MARSU Implementation Context, IBEP20, Ownable 					
L <Constructor> Public !	●	NO !			
L name Public !	!	NO !			
L symbol Public !	!	NO !			
L decimals Public !	!	NO !			
L totalSupply Public !	!	NO !			
L balanceOf Public !	!	NO !			
L allowance Public !	!	NO !			
L approve Public !	!	●	NO !		

CONTRACT ASSESSMENT

L transferFrom Public	!	● NO	!	
L increaseAllowance Public	!	● NO	!	
L decreaseAllowance Public	!	● NO	!	
L transfer Public	!	● NO	!	
L isExcludedFromReward Public	!	NO	!	
L reflectionFromToken Public	!	NO	!	
L EnableTrading External	!	● onlyOwner		
L updatedeadline External	!	● onlyOwner		
L tokenFromReflection Public	!	NO	!	
L excludeFromReward Public	!	● onlyOwner		
L includeInReward External	!	● onlyOwner		
L excludeFromFee Public	!	● onlyOwner		
L includeInFee Public	!	● onlyOwner		
L isExcludedFromFee Public	!	NO	!	
L _reflectRfi Private	🔒	●		
L _takeLiquidity Private	🔒	●		
L _takeMarketing Private	🔒	●		
L _takeOps Private	🔒	●		
L _takeDev Private	🔒	●		
L _getValues Private	🔒			
L _getTValues Private	🔒			
L _getRValues1 Private	🔒			
L _getRValues2 Private	🔒			
L _getRate Private	🔒			
L _getCurrentSupply Private	🔒			
L _approve Private	🔒	●		
L _transfer Private	🔒	●		
L _tokenTransfer Private	🔒	●		
L swapAndLiquify Private	🔒	●	lockTheSwap	
L addLiquidity Private	🔒	●		
L swapTokensForBNB Private	🔒	●		
L bulkExcludeFromFee External	!	●	onlyOwner	
L bulkIncludeInFee External	!	●	onlyOwner	
L updateMarketingWallet External	!	●	onlyOwner	
L updateDevWallet External	!	●	onlyOwner	
L updateOpsWallet External	!	●	onlyOwner	
L updateSwapTokensAtAmount External	!	●	onlyOwner	
L updateSwapEnabled External	!	●	onlyOwner	
L rescueBNB External	!	●	onlyOwner	
L rescueAnyBEP20Tokens Public	!	●	onlyOwner	
L <Receive Ether> External	!	💸	NO	!

CONTRACT ASSESSMENT

Legend

Symbol	Meaning
●	Function can modify state
\\$	Function is payable



STATIC ANALYSIS

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
MARSU.includeInReward(address) (contracts/Token.sol#404-415) has costly operations inside a loop:
  - _excluded.pop() (contracts/Token.sol#411)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#46-49) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

MARSU._rTotal (contracts/Token.sol#165) is set pre-construction with a non-constant function or state variable:
  - (MAX - (MAX % tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.20 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#126-137):
  - (success) = recipient.call{value: amount}() (contracts/Token.sol#132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#102) is not in mixedCase
Struct MARSU.valuesFromGetValues (contracts/Token.sol#202-216) is not in CapWords
Function MARSU.EnableTrading() (contracts/Token.sol#370-375) is not in mixedCase
Parameter MARSU.updatedDeadline(uint256).deadline (contracts/Token.sol#377) is not in mixedCase
Parameter MARSU.updateSwapEnabled(bool).enabled (contracts/Token.sol#796) is not in mixedCase
Parameter MARSU.rescueAnyBEP20Tokens(address,address,uint256).tokenAddr (contracts/Token.sol#808) is not in mixedCase
Parameter MARSU.rescueAnyBEP20Tokens(address,address,uint256).to (contracts/Token.sol#809) is not in mixedCase
Parameter MARSU.rescueAnyBEP20Tokens(address,address,uint256).amount (contracts/Token.sol#810) is not in mixedCase
Constant MARSU._decimals (contracts/Token.sol#161) is not in UPPER_CASE_WITH_UNDERSCORES
Variable MARSU.genesis_block (contracts/Token.sol#169) is not in mixedCase
Constant MARSU._name (contracts/Token.sol#177) is not in UPPER_CASE_WITH_UNDERSCORES
Constant MARSU._symbol (contracts/Token.sol#178) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#47)" inContext (contracts/Token.sol#41-50)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

MARSU.updateSwapTokensAtAmount(uint256) (contracts/Token.sol#788-794) uses literals with too many digits:
  - require(bool,string)(amount <= 4206900000000,Cannot set swap threshold amount higher than 1% of tokens) (contracts/Token.sol#789-792)
MARSU.slitherConstructorVariables() (contracts/Token.sol#140-820) uses literals with too many digits:
  - _tTotal = 42069000000000 * 10 ** _decimals (contracts/Token.sol#164)
MARSU.slitherConstructorVariables() (contracts/Token.sol#140-820) uses literals with too many digits:
  - swapTokensAtAmount = 420690000000 * 10 ** 9 (contracts/Token.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits

MARSU._lastSell (contracts/Token.sol#156) is never used in MARSU (contracts/Token.sol#140-820)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

MARSU._tTotal (contracts/Token.sol#164) should be constant
MARSU.deadWallet (contracts/Token.sol#172) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

MARSU.pair (contracts/Token.sol#159) should be immutable
MARSU.router (contracts/Token.sol#158) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Static Analysis

an static analysis of the code were performed using slither. No issues were found



FUNCTIONAL TESTING

Router (PCS V2):

0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):

<https://testnet.bscscan.com/tx/0xdfd4f148cc87e8f446915375c11cf3ff71a071cbbfd4f48b61a6bab80d8133a5>

2- Buying (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x76a99b4ed95af6791a801d5836c40225a12017cebeec6cf4d9475eb7d7a2fef0>

3- Selling (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x21ec2cd2879ded1277e7ffb980ef7df53db12956d2dff0e5aea3a330ffa5532>

4- Transferring 0% tax) (passed):

<https://testnet.bscscan.com/tx/0xffd6e789b130340fcebe2dfd14ffc7fb125ac6750b94c1b4b2fecbd0a052919b>

5- Buying when not excluded from fees (8% tax) (passed):

<https://testnet.bscscan.com/tx/0x91feae0e3405aa3c45ad94015fb2f444c199570f6b1891659632bf3d120d0a70>

6- Selling when not excluded from fees (8% tax) (passed):

<https://testnet.bscscan.com/tx/0x052cb01bf5815a13f65fd8f93803cefa12be3e897eba4959059419d3201f08c6>



FUNCTIONAL TESTING

7- Transferring when not excluded from fees (8% tax) (passed):

<https://testnet.bscscan.com/tx/0xa2e058c876a3984e606f5cad449869ec33a2e19024c53490624ba2bde48c6a3b>

8- Internal swap(passed):

<https://testnet.bscscan.com/tx/0xa2e058c876a3984e606f5cad449869ec33a2e19024c53490624ba2bde48c6a3b>



MANUAL TESTING

Logical – zero swapTokensAtAmount can disable sell/transfers

Severity: **Critical**

function: updateSwapTokensAtAmount

Status: Open

Overview:

Setting swapTokensAtAmount to 0 can disable sell and transfer transactions for regular wallets (non whitelisted), this is because even if swapTokensAtAmount is set to 0, internal swap is still performed and reverts the transaction in attempt to swap 0 tokens for bnb.

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {  
    require(amount <= 4206900000000, "Cannot set swap threshold amount higher than 1% of  
tokens");  
    swapTokensAtAmount = amount * 10**_decimals;  
}
```

Suggestion

Ensure that swapTokensAtAmount is always greater than a reasonable minimum value:

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {  
    require(amount <= 4206900000000, "Cannot set swap threshold amount higher than 1% of  
tokens");  
    require(amount >= 4206900000, "Cannot set swap threshold amount higher than  
0.0001% of tokens");  
    swapTokensAtAmount = amount * 10 ** _decimals;  
}
```



FUNCTIONAL TESTING

Centralization – Trades must be enabled

Severity: **Informational**

function: EnableTrading

Status: **Resolved (owned by safu dev)**

Overview:

The smart contract owner must enable trades for holders. If trading remain disabled, no one would be able to buy/sell/transfer tokens.

```
function EnableTrading() external onlyOwner {  
    require(!tradingEnabled, "Cannot re-enable trading");  
    tradingEnabled = true;  
    swapEnabled = true;  
    genesis_block = block.number;  
}
```

Suggestion

To mitigate this centralization issue, we propose the following options:

1. Renounce Ownership: Consider relinquishing control of the smart contract by renouncing ownership. This would remove the ability for a single entity to manipulate the router, reducing centralization risks.
2. Multi-signature Wallet: Transfer ownership to a multi-signature wallet. This would require multiple approvals for any changes to the mainRouter, adding an additional layer of security and reducing the centralization risk.
3. Transfer ownership to a trusted and valid 3rd party in order to guarantee enabling of the trades



DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



ABOUT AUDITACE

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



<https://auditace.tech/>



https://t.me/Audit_Ace



https://twitter.com/auditace_



<https://github.com/Audit-Ace>
