auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

SAFEMOON

DATED : 9 July 23'

HIGH RISK FINDING

Centralization – Enabling Trades

Severity: High

function: enableTrading

Status: Not Resolved

Overview:

Owner of the contract must enable trades manually for investors, otherwise no one would be able to buy/sell/transfer their tokens.

```
function enableTrading() external onlyOwner {
    require(!tradingActive, "Cannot enable trading again");
    tradingActive = true;
    swapEnabled = true;
    tradingBlock = block.number;
}
```

Suggestion

Its suggested to either enable trades prior to presale, or transfer ownership of the contract to a certified pinsksale safu developer to guearanee enabling of trades.

AUDIT SUMMARY

Project name – SAFEMOON

Date: 9 July, 2023

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

Audit Status: **Passed with High Risk**

Issues Found

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

USED TOOLS

Tools:

1- Manual Review:
A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC 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/0x91f5a218cEa22CD3Cc3A11d812A3Dd135592368C

Token Information

Token Name : SafeMoon2.0

Token Symbol: SAFEMOON

Decimals: 18

Token Supply: 1,000,000

Token Address:
0xae9dFbdDFEbbBc6B42293B922EE50BCC1539616d

Checksum:
14f7b4fabe9e0ff89107233d95622b89e90f4699

Owner:
0xcCA63F257b98Fb36c52D677865888142AF94B015
(at time of writing the audit)

Deployer:
0xcCA63F257b98Fb36c52D677865888142AF94B015

TOKEN OVERVIEW

Fees:

Buy Fees: 0-3%

Sell Fees: 0-3%

Transfer Fees: 0%

Fees Privilege: Owner

Ownership: owned

Minting: none

Max Tx Amount/ Max Wallet Amount: Yes

Blacklist: No

Other Privileges: - enabling trades

- Initial distribution of the tokens

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 isexercised 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

- [x] Return values of low-level calls
- [x] Private modifier
- [x] Multiple Sends
- [x] Using Suicide
- [x] Gas Limitand Loops
- [x] Address hardcoded
- [x] Exception Disorder
- [x] Using inline assembly
- [x] Divide before multiply
- [x] Missing Zero Address Validation
- [x] Compiler version not fixed
- [x] **Gasless Send**
- [x] Using block.timestamp
- [x] Re-entrancy
- [x] Tautology or contradiction
- [x] Timestamp Dependence
- [x] Revert/require functions
- [x] Use of tx.origin
- [x] Integer overflow/underflow
- [x] Dangerous strict equalities
- [x] Using SHA3
- [x] Using throw

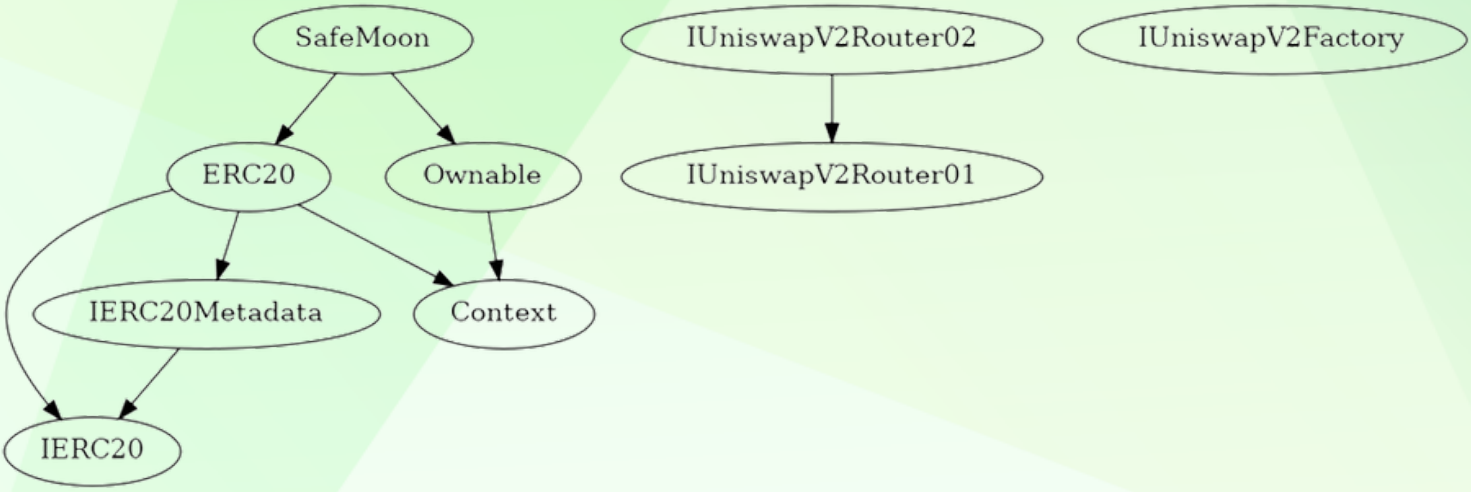
CLASSIFICATION OF RISK

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	1
Medium-Risk	1
Low-Risk	0
Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to update buy/sell fees within 0-3% (0% transfer fees)
- Owner is not able to blacklist an address
- Owner is not able to disable buy/sell/transfers
- Owner is not able to set max wallet limit and minimum wallet limits
- Owner is not able to mint new tokens
- **Owner must enable trades manually**

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
└	totalSupply	External ❗		NO ❗
└	balanceOf	External ❗		NO ❗
└	transfer	External ❗	🛑	NO ❗
└	allowance	External ❗		NO ❗
└	approve	External ❗	🛑	NO ❗
└	transferFrom	External ❗	🛑	NO ❗
IERC20Metadata	Interface	IERC20		
└	name	External ❗		NO ❗
└	symbol	External ❗		NO ❗
└	decimals	External ❗		NO ❗
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
ERC20	Implementation	Context, IERC20, IERC20Metadata		
└	<Constructor>	Public ❗	🛑	NO ❗
└	name	Public ❗		NO ❗
└	symbol	Public ❗		NO ❗
└	decimals	Public ❗		NO ❗
└	totalSupply	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	transfer	Public ❗	🛑	NO ❗
└	allowance	Public ❗		NO ❗
└	approve	Public ❗	🛑	NO ❗
└	transferFrom	Public ❗	🛑	NO ❗
└	increaseAllowance	Public ❗	🛑	NO ❗
└	decreaseAllowance	Public ❗	🛑	NO ❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_spendAllowance	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	Internal 🔒	🛑	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗

CONTRACT ASSESMENT

└	owner	Public ❗		NO ❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
IUniswapV2Router01	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	removeLiquidity	External ❗	🛑	NO ❗
└	removeLiquidityETH	External ❗	🛑	NO ❗
└	removeLiquidityWithPermit	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermit	External ❗	🛑	NO ❗
└	swapExactTokensForTokens	External ❗	🛑	NO ❗
└	swapTokensForExactTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokens	External ❗	💵	NO ❗
└	swapTokensForExactETH	External ❗	🛑	NO ❗
└	swapExactTokensForETH	External ❗	🛑	NO ❗
└	swapETHForExactTokens	External ❗	💵	NO ❗
└	quote	External ❗		NO ❗
└	getAmountOut	External ❗		NO ❗
└	getAmountIn	External ❗		NO ❗
└	getAmountsOut	External ❗		NO ❗
└	getAmountsIn	External ❗		NO ❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO ❗
└	feeToSetter	External ❗		NO ❗
└	getPair	External ❗		NO ❗
└	allPairs	External ❗		NO ❗
└	allPairsLength	External ❗		NO ❗
└	createPair	External ❗	🛑	NO ❗
└	setFeeTo	External ❗	🛑	NO ❗

CONTRACT ASSESMENT

└	setFeeToSetter	External ❗	🛑	NO ❗
SafeMoon	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🛑	ERC20
└	airdropToWallets	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	swapBack	Private 🔐	🛑	
└	swapTokensForBNB	Internal 🔒	🛑	
└	safeTransferBNB	Internal 🔒	🛑	
└	addLiquidity	Private 🔐	🛑	
└	excludeFromFees	Public ❗	🛑	onlyOwner
└	_setAutomatedMarketMakerPair	Private 🔐	🛑	
└	setAutomatedMarketMakerPair	External ❗	🛑	onlyOwner
└	updateBuyFees	External ❗	🛑	onlyOwner
└	updateSellFees	External ❗	🛑	onlyOwner
└	updateDevelopmentWallet	External ❗	🛑	onlyOwner
└	updateTeamWallet	External ❗	🛑	onlyOwner
└	updateLiquidityWallet	External ❗	🛑	onlyOwner
└	updateSwapTokensAtAmount	External ❗	🛑	onlyOwner

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Different versions of Solidity are used:
        - Version used: ['>=0.4.22<0.9.0', '>=0.5.0', '>=0.6.2', '^0.8.0', '^0.8.17']
        - >=0.4.22<0.9.0 (contracts/Token.sol#809)
        - >=0.5.0 (contracts/Token.sol#786)
        - >=0.6.2 (contracts/Token.sol#595)
        - >=0.6.2 (contracts/Token.sol#735)
        - ^0.8.0 (contracts/Token.sol#99)
        - ^0.8.0 (contracts/Token.sol#126)
        - ^0.8.0 (contracts/Token.sol#151)
        - ^0.8.0 (contracts/Token.sol#514)
        - ^0.8.17 (contracts/Token.sol#20)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Context._msgData() (contracts/Token.sol#143-145) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#421-437) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version^0.8.17 (contracts/Token.sol#20) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
Pragma version^0.8.0 (contracts/Token.sol#99) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#126) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#151) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#514) allows old versions
Pragma version>=0.6.2 (contracts/Token.sol#595) allows old versions
Pragma version>=0.6.2 (contracts/Token.sol#735) allows old versions
Pragma version>=0.5.0 (contracts/Token.sol#786) allows old versions
Pragma version>=0.4.22<0.9.0 (contracts/Token.sol#809) is too complex
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 SafeMoon.safeTransferBNB(address,uint256) (contracts/Token.sol#995-998):
        - (success) = to.call{value: value}(new bytes(0)) (contracts/Token.sol#996)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Router01.WETH() (contracts/Token.sol#600) is not in mixedCase
Event SafeMoondevelopmentWalletUpdated(address,address) (contracts/Token.sol#843) is not in CapWords
Parameter SafeMoon.updateBuyFees(uint256,uint256,uint256)._developmentFee (contracts/Token.sol#1032) is not in mixedCase
Parameter SafeMoon.updateBuyFees(uint256,uint256,uint256)._liquidityFee (contracts/Token.sol#1032) is not in mixedCase
Parameter SafeMoon.updateBuyFees(uint256,uint256,uint256)._teamFee (contracts/Token.sol#1032) is not in mixedCase
Parameter SafeMoon.updateSellFees(uint256,uint256)._developmentFee (contracts/Token.sol#1040) is not in mixedCase
Parameter SafeMoon.updateSellFees(uint256,uint256)._liquidityFee (contracts/Token.sol#1040) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#605) is too s
imilar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#606)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SafeMoon.constructor(uint256) (contracts/Token.sol#847-873) uses literals with too many digits:
        - swapTokensAtAmount = (initialSupply * 1) / 100000 (contracts/Token.sol#866)
SafeMoon.updateSwapTokensAtAmount(uint256) (contracts/Token.sol#1065-1070) uses literals with too many digits:
        - require(bool,string)(newAmount >= (totalSupply() * 1) / 100000,Swap amount cannot be lower than 0.001% total supply.) (contracts/Token.sol#1066)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xd0360f0168e95d49b430b4878d779cdfb237299c651ca9b6398bf704edf209ca

2- Buying when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x3be7b48e68e5bccb7c622dc8f51a959b2fe5e440703d50f1560822112a053ec2

3- Selling when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x0e9c410f31afe91a9522b0f6d2fc9fc532396a38dcab3c703bcb14325b2dcf64

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xfe67efdd9c38d58e59cba939c7f187da16b6401cacc3f79c54cca83116d5b309

5- Buying(0-3% tax) (passed):
https://testnet.bscscan.com/tx/0x44acacfe7751120d1627c9d8206151a5001d5b82c20795e78645e01f64e80e42

6- Selling (0-3% tax) (passed):
https://testnet.bscscan.com/tx/0x066712bc55291e1eabf9c46052e8b829d5770b5cea376df3cfedfb526930da6f

FUNCTIONAL TESTING

4- Transferring (0% tax) (passed):

https://testnet.bscscan.com/tx/0x3fd442bca2f2293afd3620ffea0025a2b97509850c2c1bd54c9b0f5c2114d212

4- Internal swap (ETH sent to marketing wallet | auto-liquidity) (passed):

https://testnet.bscscan.com/tx/0x066712bc55291e1eabf9c46052e8b829d5770b5cea376df3cfedfb526930da6f

FUNCTIONAL TESTING

Centralization – Enabling Trades

Severity: High

function: enableTrading

Status: Not Resolved

Overview:

Owner of the contract must enable trades manually for investors, otherwise no one would be able to buy/sell/transfer their tokens.

```
function enableTrading() external onlyOwner {
    require(!tradingActive, "Cannot enable trading again");
    tradingActive = true;
    swapEnabled = true;
    tradingBlock = block.number;
}
```

Suggestion

Its suggested to either enable trades prior to presale, or transfer ownership of the contract to a certified pinsksale safu developer to guearanee enabling of trades.

FUNCTIONAL TESTING

Centralization – EOA receiving LP tokens

Severity: Medium

function: addLiquidity

Status: Not Resolved

Overview:

an EOA (externally owned account) is received LP tokens generated from auto-liquidity.

```
uniswapV2Router.addLiquidityETH{value: bnbAmount}(
    address(this),
    tokenAmount,
    0, // slippage is unavoidable
    0, // slippage is unavoidable
    liquidityAddress,
    block.timestamp
);
```

Suggestion

Its suggested to burn (sending LP tokens to dead address) or lock new LP tokens.

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 specializes 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