RugFreeCoins Audit

Socaverse Token

Smart Contract Security Audit

January 25, 2022

Contents

Audit details

Audited project

Socaverse Token

Contract Address

0x57Af121A8dDb1F9A8FafcF3229c92bF2856A8a29

Client contact

Socaverse Team

Blockchain

Binance smart chain

Project website

https://www.socaverse.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 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 Rugfreecoins and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Rugfreecoins) owe no duty of care towards you or any other person, nor does Rugfreecoins 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 Rugfreecoins 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, Rugfreecoins hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Rugfreecoins, 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.

Background

Rugfreecoins was commissioned by Socaverse Token to perform an audit of the smart contract.

https://bscscan.com/token/0x57Af121A8dDb1F9A8FafcF3229c92bF2856A8a29

The focus of this audit is to verify that the smart contract is secure, resilient and working according to the specifications.

The information in this report should be used to understand the risk exposure of the smart contract, project feasibility, long term sustainability and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

About the project

Socaverse is a token built on the Binance Smart Chain that is with an innovative investment use case the main purpose of which is to seek out constant revenue sources, and heading towards building even greater Community, more games and crypto-related features that engage users and token holders. They are building a full-stack Metaverse company that enables players to enter the ecosystem and be rewarded from their NFTs with play-to-earn games. Each transaction, purchase and sale incurs 8% fee.

Socaverse goal is to become the crypto gateway for non-crypto users and aims to build a metaverse that connects the physical world to the digital world. Blockchain is their tool for building a gateway between two massive industries, and SOCA is a gate to enter the Metaverse.

Features

- The **Socaverse Token rewards** will be distributed among every holder proportional to how many tokens each individual holds in values of **2% when buying and selling**.

- The **sustainability fee of 2% when buying and selling for dev** is what allows Socaverse to hold the aforementioned promise. Tokens will be swapped into BNB and will be sent to a Dev wallet per transaction. This way, Socaverse will have enough funds to promote the coin and spend for future development without selling tokens as the traditional way.

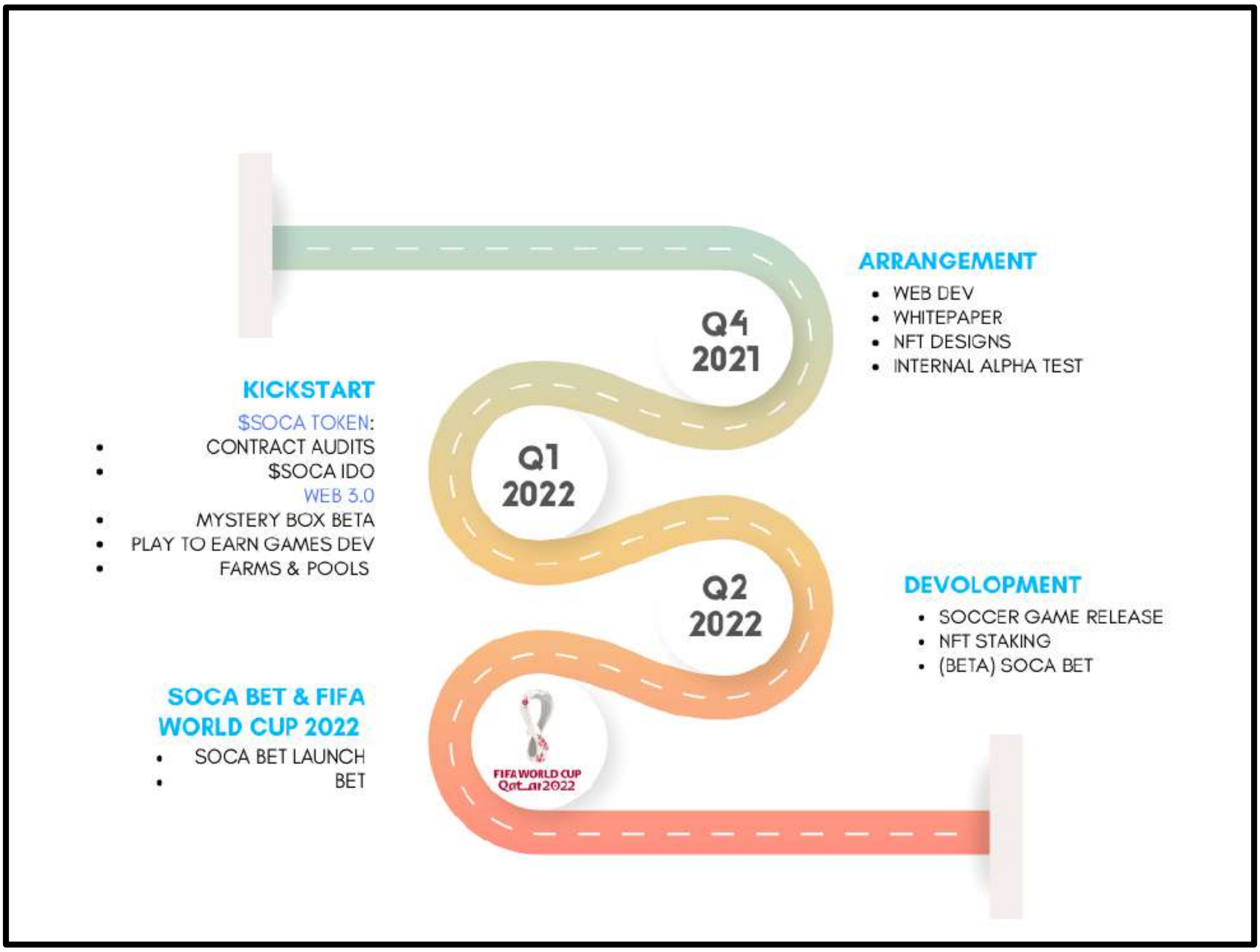
- The additional component included under the sustainability section is a **liquidity fee of 4% when buying and selling**, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

Tokenomics

8% fee when buying and selling

- 2% of trade goes to holders' pockets in Socaverse tokens.
- 4% of trade goes to the liquidity pool.
- 2% of trade goes to the Dev wallet.

Roadmap



Q4 2021

ARRANGEMENT

- WEB DEV
- WHITEPAPER
- NFT DESIGNS
- INTERNAL ALPHA TEST

Q1 2022

KICKSTART

$SOCA TOKEN:

- CONTRACT AUDITS
- $SOCA IDO

WEB 3.0

- MYSTERY BOX BETA
- PLAY TO EARN GAMES DEV
- FARMS & POOLS

Q2 2022

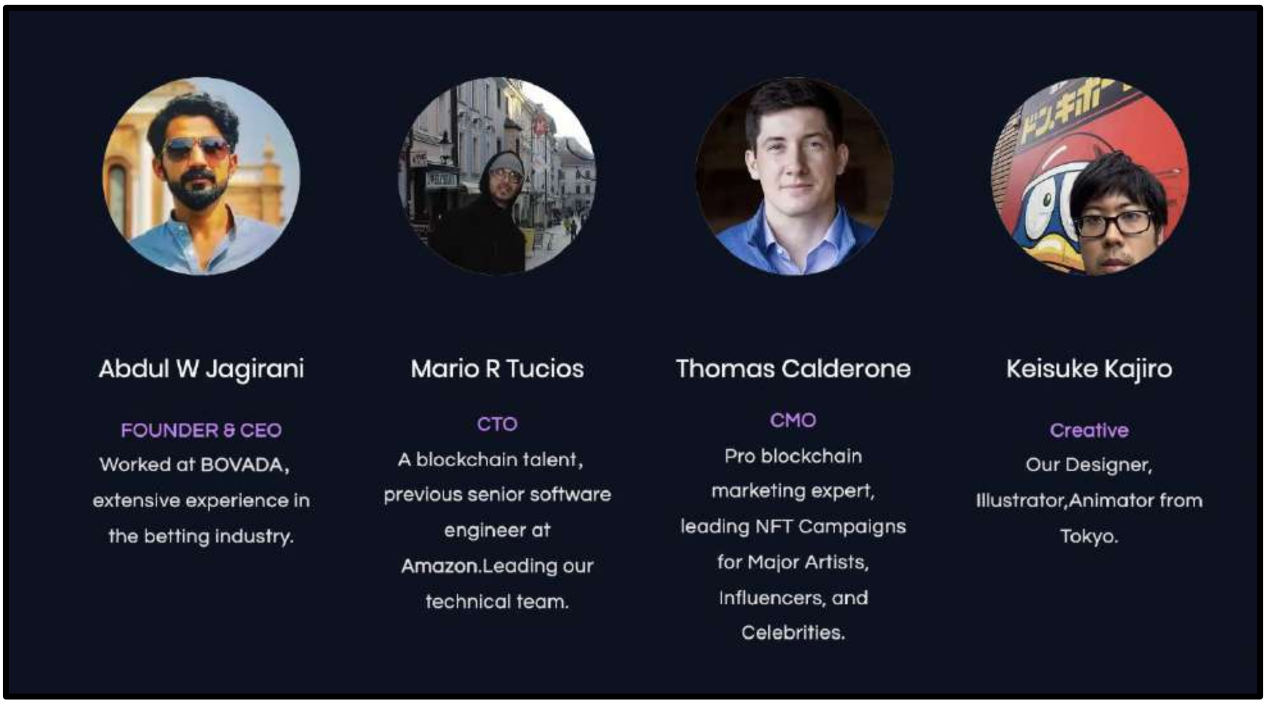
DEVOLOPMENT

- SOCCER GAME RELEASE
- NFT STAKING
- (BETA) SOCA BET

SOCA BET & FIFA WORLD CUP 2022

- SOCA BET LAUNCH
- BET

The Team



Abdul W Jagirani

FOUNDER & CEO

Worked at BOVADA, extensive experience in the betting industry.

Mario R Tucios

CTO

A blockchain talent, previous senior software engineer at Amazon.Leading our technical team.

Thomas Calderone

CMO

Pro blockchain marketing expert, leading NFT Campaigns for Major Artists, Influencers, and Celebrities.

Keisuke Kajiro

Creative

Our Designer, Illustrator,Animator from Tokyo.

Target market and the concept

Target market

- Anyone who's interested in the Crypto space with long-term investment plans.
- Anyone who's ready to earn a passive income by holding tokens.
- Anyone who's interested in trading tokens.
- Anyone who's interested in collecting NFTs or trading NFTs.
- Anyone who's interested in taking part with the betting and play to earn game that will be released in the future.
- Anyone who's interested in taking part with the NFT staking.
- Anyone who's interested in taking part with the future plans of the Socaverse token.
- Anyone who's interested in making financial transactions with any other party using Socaverse as the currency.

Core concept

The Socaverse reward system

2% of each transaction when buying and selling gets converted to tokens and is split amongst all holders. Holders will be eligible to receive tokens in each transaction and rewards are proportional to how many tokens each individual holds.

Sustainable mechanism

The sustainability fee of 2% when buying and selling for dev is what allows Socaverse to promote the token and use funds to further the development of the platform. Tokens will be swapped into BNB and will be sent to a marketing wallet per transaction. This way, Socaverse will have access to the funds without selling tokens as the traditional way, which will enable them to consume funds without hurting the project.

The liquidity fee of 4% when buying and selling, which is a redistribution mechanism that ensures the trading pool always has sufficient liquidity.

The use cases

Socaverse aims to build a vertical metaverse that attracts non-crypto soccer lovers to play and earn through multiple revenue streams. In SOCA, players can purchase different types and levels of mystery boxes at different prices. Each mystery box is limited and has a certain probability of drawing a frictional NFT card. SOCA guarantees users a higher value NFT than the cost of the mystery box itself. All information regarding mystery boxes will be transparent through the power of the blockchain. The odds of winning will be 100% secured by the third-party Chainlink VRF via blockchain. The probability and the purchase history of each mystery box will be recorded and made searchable on the ledger.

Other than mystery boxes, SOCA provides card attributes for users to gain more leverage with their NFTs. In the Fragment Refiner, users can upgrade their fractional NFTs to higher-value fragments. In the Forging Factory, users can forge their fractional NFTs into a single NFT and redeem the corresponding merchandise. Lastly, in the NFT Morph Factory, users can merge NFTs into new random NFTs. In the marketplace, users can buy or sell their fractional NFTs. By collecting fractional NFT cards, users will be able to unlock rare fractional NFT cards and get access to purchase an exclusive mystery box.

In the future, SOCA will develop games to keep attracting users to the platform. In the SOCA open-world metaverse, users can purchase NFTs to decorate their avatars. By playing games and completing tasks, users can earn rewards daily.

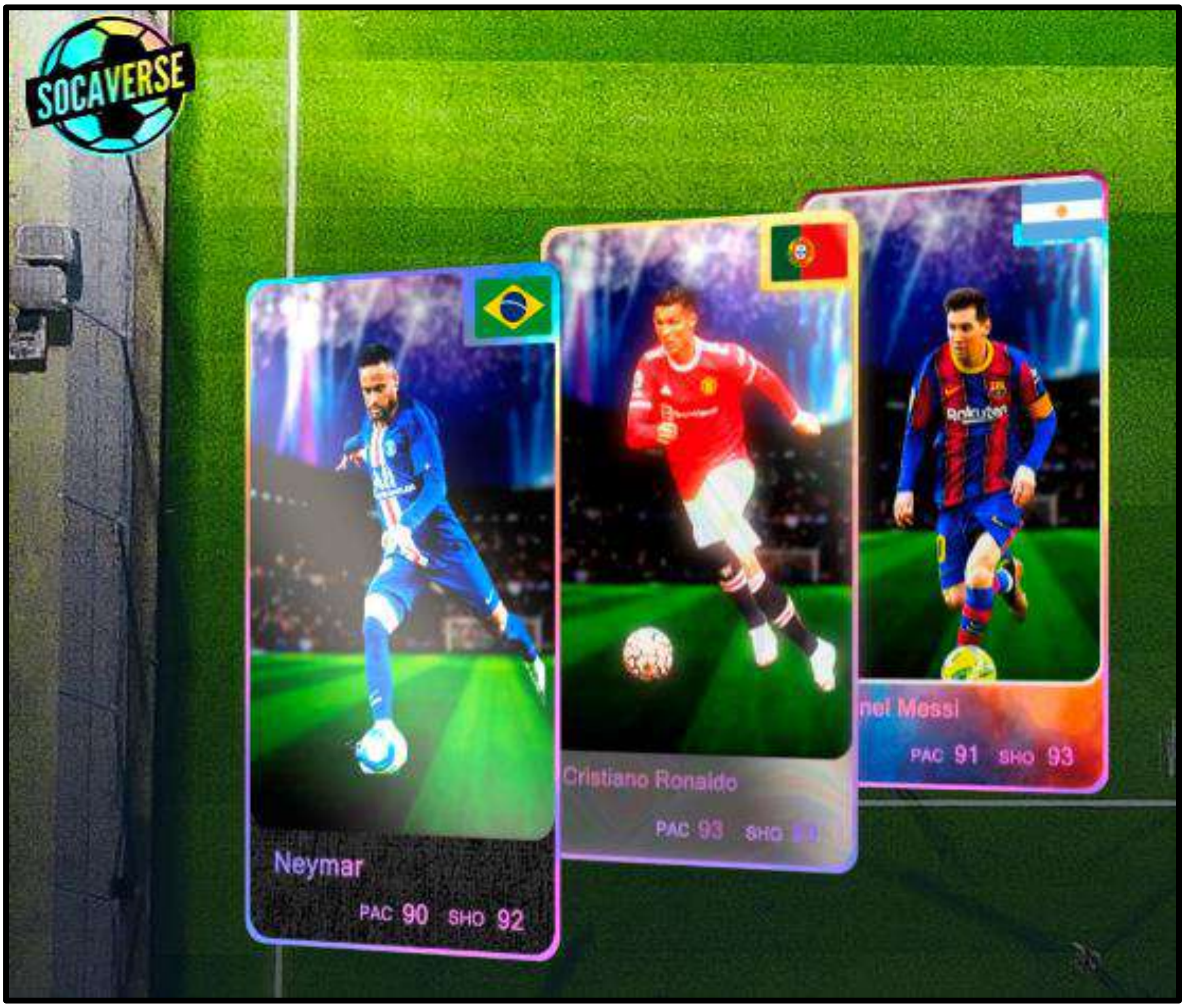
SOCA will develop more crypto native features. Users can stake (deposit) mystery boxes, merchandise, and fragments (NFTs). They will earn yields and other benefits with more significant staking amounts, as well as deposit their NFTs for collateral loans. SOCA will also allow users to use SOCA Pay to purchase merchandise from our partners by using their fractional NFTs.

Furthermore, users can deposit real-world products or collections for authentication. Our certified partners will authenticate the items. Once the authentication process is completed, SOCA will create digital NFTs of the real-world assets, and these NFTs will be available for users to use in SOCA.

By establishing a Decentralized Autonomous Organization (DAO), SOCA allows users to vote on system upgrades, select merchandise, purchase the newest fashion drops, participate in special events, and access early testing.

In the future, Socaverse plan to build a cross-chain marketplace for authenticating and trading physical NFTs.

NFT cards



NFT Cards can be obtained by opening the Soca Box or purchasing/swapping from a secondary marketplace. The cards will be divided into 3 levels: Pros (White), Stars (Blue), and Legends (Red), Each card will be generated with the winning merchandise on a random background, and a random frame corresponding to the 3 levels of rarity. The cards will have different effects on our gaming processes such as boosting the success rate of upgrading fragments. Each card will have special attributes, such as boosting mining speed, boosting game success rate, dropping random fragments, airdropping $SOCA tokens, and more. Users can collect cards to exchange for rare merchandise.

Potential to grow with score points

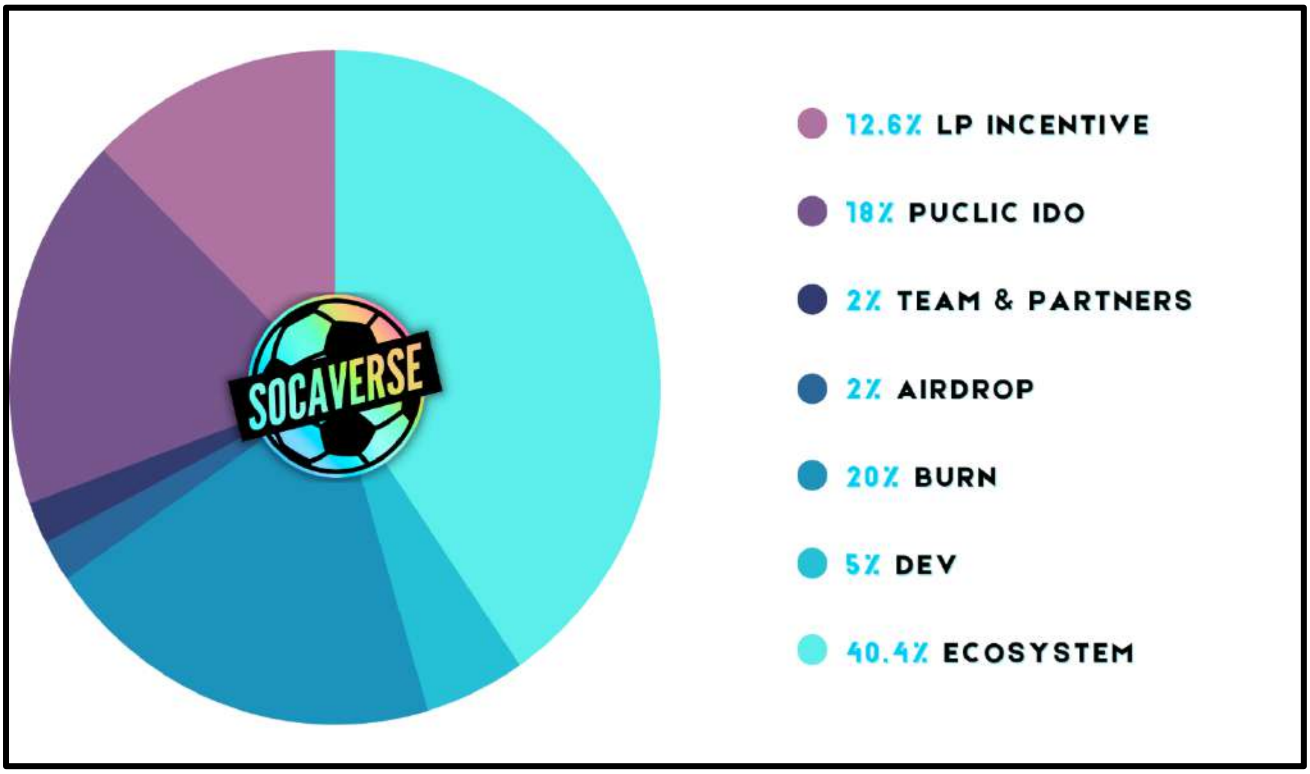
1.	Project efficiency	9/10
2.	Project uniqueness	9/10
3	Information quality	10/10
4	Service quality	10/10
5	System quality	10/10
6	Impact on the community	10/10
7	Impact on the business	10/10
8	Preparing for the future	10/10
Total Points		**9.75/10**

Contract details

Token contract details for 25th January 2022

Contract name	Socaverse
Contract address	0x57Af121A8dDb1F9A8FafcF3229c92bF2856A8a29
Token supply	10,000,000,000,000
Token ticker	SOCA
Decimals	9
Token holders	1
Transaction count	1
Dev address	0x0c0886a8db6c57a4a97057517482856bb09abf35
Marketing address	0x0c0886a8db6c57a4a97057517482856bb09abf35
Contract deployer address	0x21BbEbD8A251c1b7F0e4464bfe160Ca57077f818
Contract’s current owner address	0x21bbebd8a251c1b7f0e4464bfe160ca57077f818

Token distribution



Contract code function details

No	Category	Item	Result
1	Coding conventions	BRC20 Token standards	pass
		compile errors	pass
		Compiler version security	pass
		visibility specifiers	pass
		Gas consumption	pass
		SafeMath features	pass
		Fallback usage	pass
		tx.origin usage	pass
		deprecated items	pass
		Redundant code	pass
		Overriding variables	pass
2	Function call audit	Authorization of function call	pass
		Low level function (call/delegate call) security	pass
		Returned value security	pass
		Selfdestruct function security	pass
3	Business security	Access control of owners	pass
		Business logics	pass
		Business implementations	pass
4	Integer overflow/underflow		pass
5	Reentrancy		pass
6	Exceptional reachable state		pass
7	Transaction ordering dependence		pass
8	Block properties dependence		pass
9	Pseudo random number generator (PRNG)		pass
10	DoS (Denial of Service)		pass
11	Token vesting implementation		pass
12	Fake deposit		pass
13	Event security		pass

Contract description table

Below table represents the summary of the contracts and methods in the token contract. We scanned the whole contract and listed down all the Interfaces, functions and implementations with its visibility and mutability.

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❗
Context	**Implementation**			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		

Ownable	**Implementation**	**Context**		
└		Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
IFactory	**Interface**			
└	createPair	External ❗	🛑	NO❗
IRouter	**Interface**			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗

Socaverse	**Implementation**	**Context, IERC20, Ownable**		
└		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❗
└	isExcludedFromReward	Public ❗		NO❗
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	Public ❗	🛑	onlyOwner

└	includeInReward	External ❗	🛑	onlyOwner
└	excludeFromFee	Public ❗	🛑	onlyOwner
└	includeInFee	Public ❗	🛑	onlyOwner
└	isExcludedFromFee	Public ❗		NO❗
└	setTaxes	Public ❗	🛑	onlyOwner
└	_reflectRfi	Private 🔐	🛑	
└	_takeLiquidity	Private 🔐	🛑	
└	_takeMarketing	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_approve	Private 🔐	🛑	

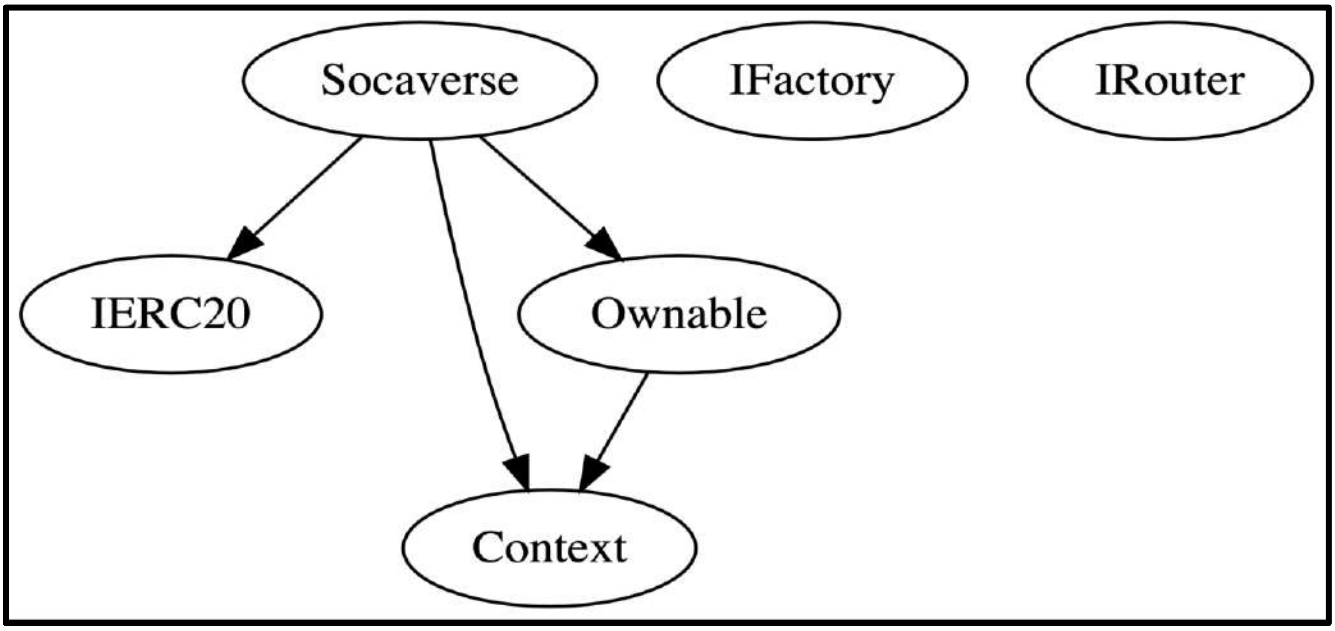
└	_transfer	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	addLiquidity	Private 🔐	🛑	
└	swapTokensForBNB	Private 🔐	🛑	
└	updateMarketingWallet	External ❗	🛑	onlyOwner
└	updateMaxWalletBalance	External ❗	🛑	onlyOwner
└	updatMaxBuyAmt	External ❗	🛑	onlyOwner
└	updateSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	updateSwapEnabled	External ❗	🛑	onlyOwner
└	setAntibot	External ❗	🛑	onlyOwner
└	bulkAntiBot	External ❗	🛑	onlyOwner
└	updateRouterAndPair	External ❗	🛑	onlyOwner

└	isBot	Public ❗		NO❗
└	rescueBNB	External ❗	🛑	onlyOwner
└	rescueAnyBEP20Tokens	Public ❗	🛑	onlyOwner
└		External ❗	💵	NO❗

Legend

Symbol	Meaning
🛑	**Function can modify state**
💵	**Function is payable**

Inheritance Hierarchy



Security issue checking status

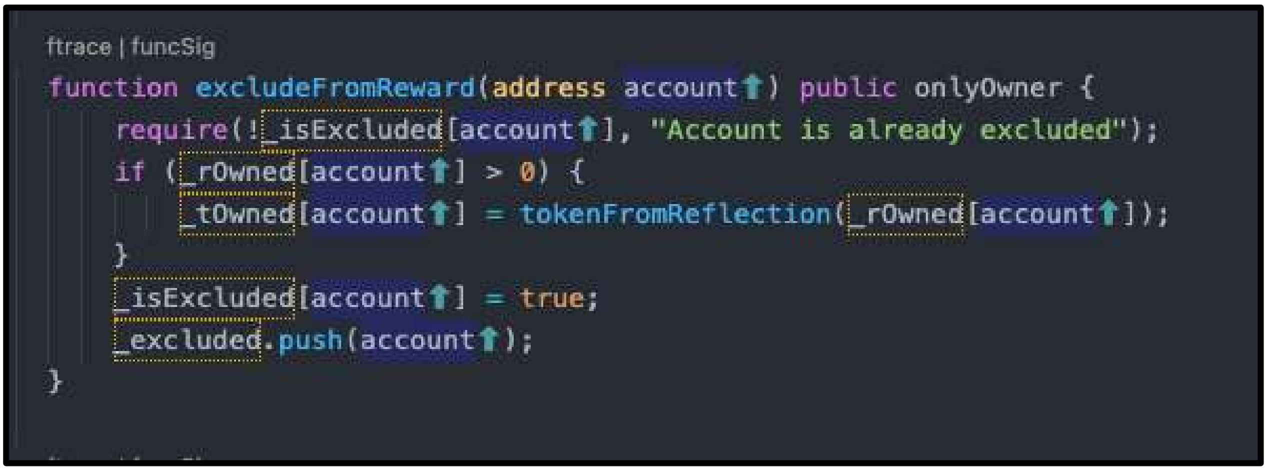
- **High severity issues**
  No high severity issues found.

- **Medium severity issues**
  No medium severity issues found.

- **Low severity issues**
  No low severity issues found.

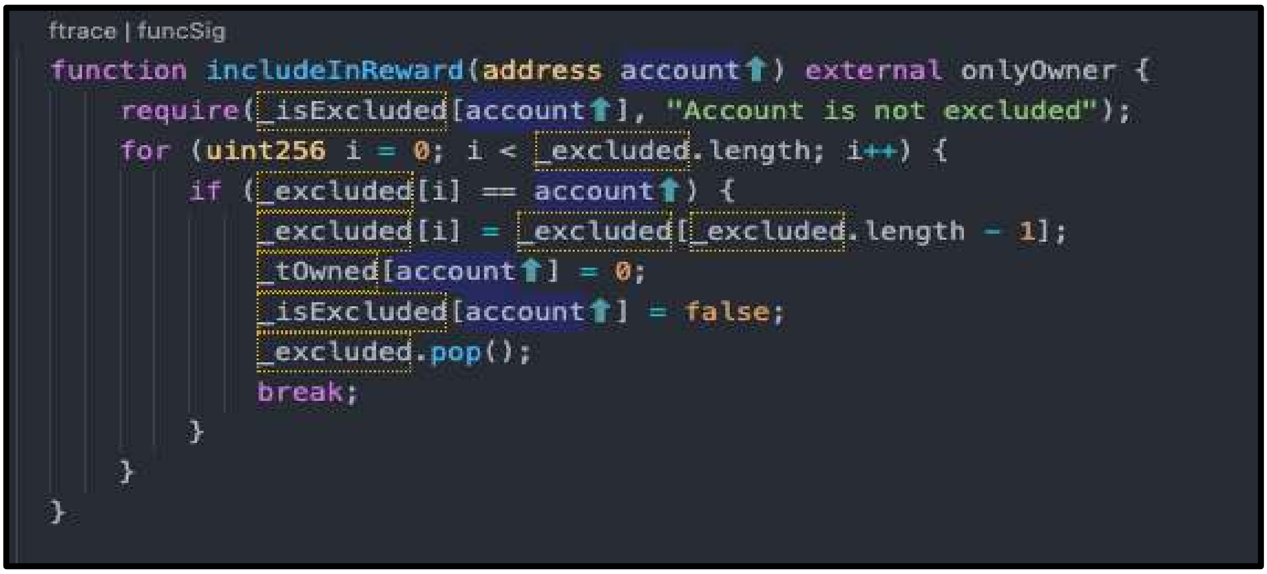
Owner privileges

- The owner can exclude wallets from rewards.



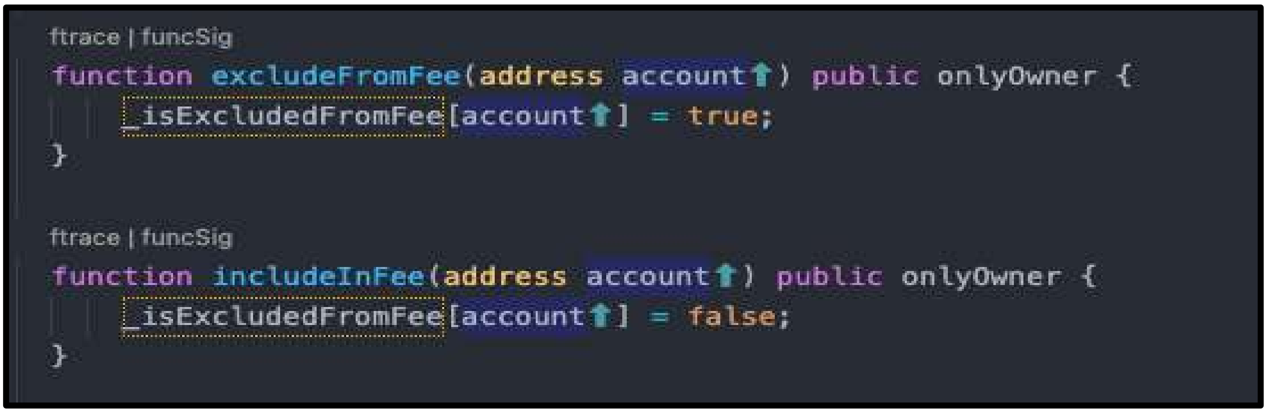
```
ftrace | funcSig
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcluded[account], "Account is already excluded");
    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}
```

- The owner can include wallets from rewards.



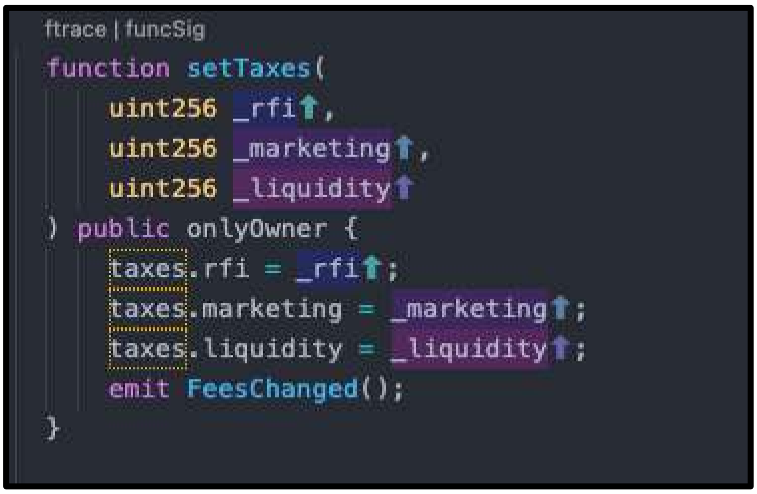
```
ftrace | funcSig
function includeInReward(address account) external onlyOwner {
    require(_isExcluded[account], "Account is not excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The owner can include/exclude wallets from fee.



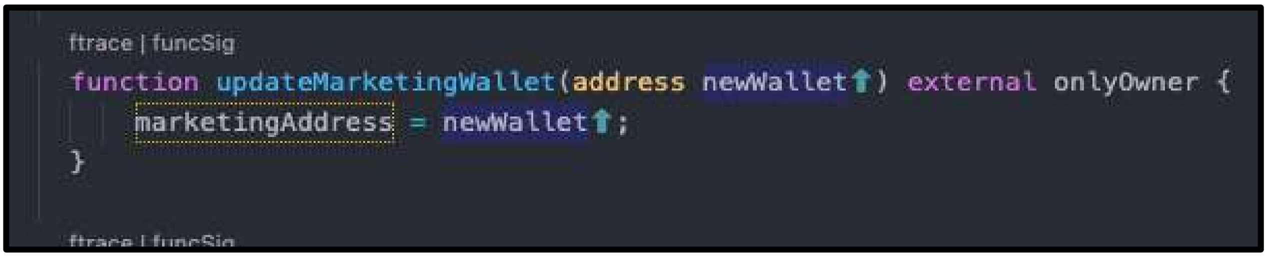
```
ftrace | funcSig
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

ftrace | funcSig
function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

- The owner can change all tax fees.



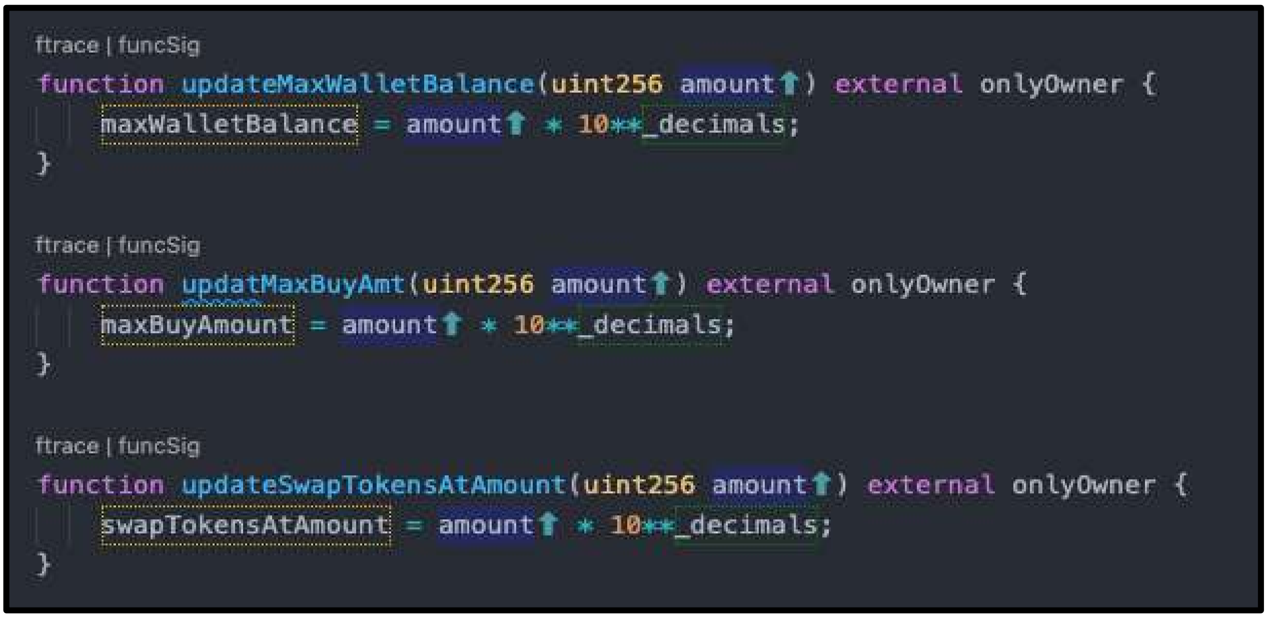
```
ftrace | funcSig
function setTaxes(
        uint256 _rfi,
        uint256 _marketing,
        uint256 _liquidity
) public onlyOwner {
        taxes.rfi = _rfi;
        taxes.marketing = _marketing;
        taxes.liquidity = _liquidity;
        emit FeesChanged();
}
```

- The owner can update the marketing wallet.



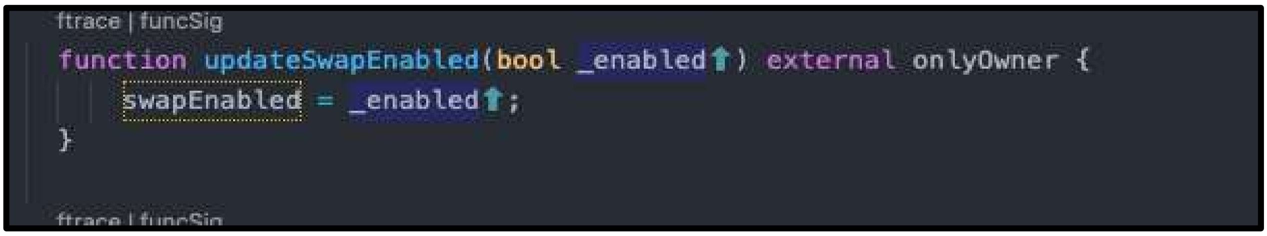
```
ftrace | funcSig
function updateMarketingWallet(address newWallet) external onlyOwner {
    marketingAddress = newWallet;
}

ftrace | funcSig
```

- The owner can change max wallet token,max buy amount and token swap point.



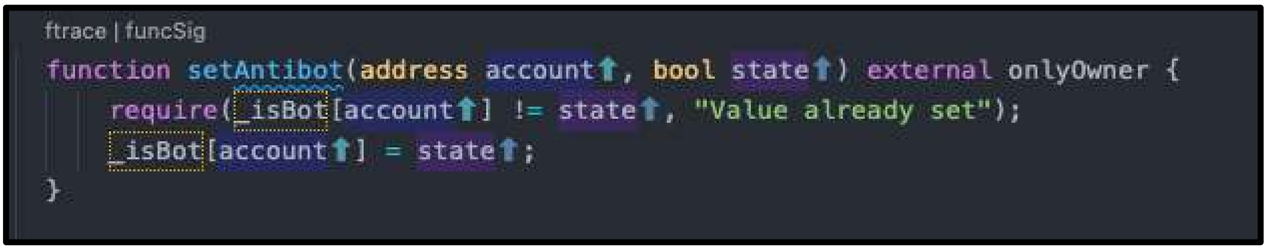
```
ftrace | funcSig
function updateMaxWalletBalance(uint256 amount) external onlyOwner {
    maxWalletBalance = amount * 10**_decimals;
}

ftrace | funcSig
function updatMaxBuyAmt(uint256 amount) external onlyOwner {
    maxBuyAmount = amount * 10**_decimals;
}

ftrace | funcSig
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
    swapTokensAtAmount = amount * 10**_decimals;
}
```

- The owner can enable/disable swaping.



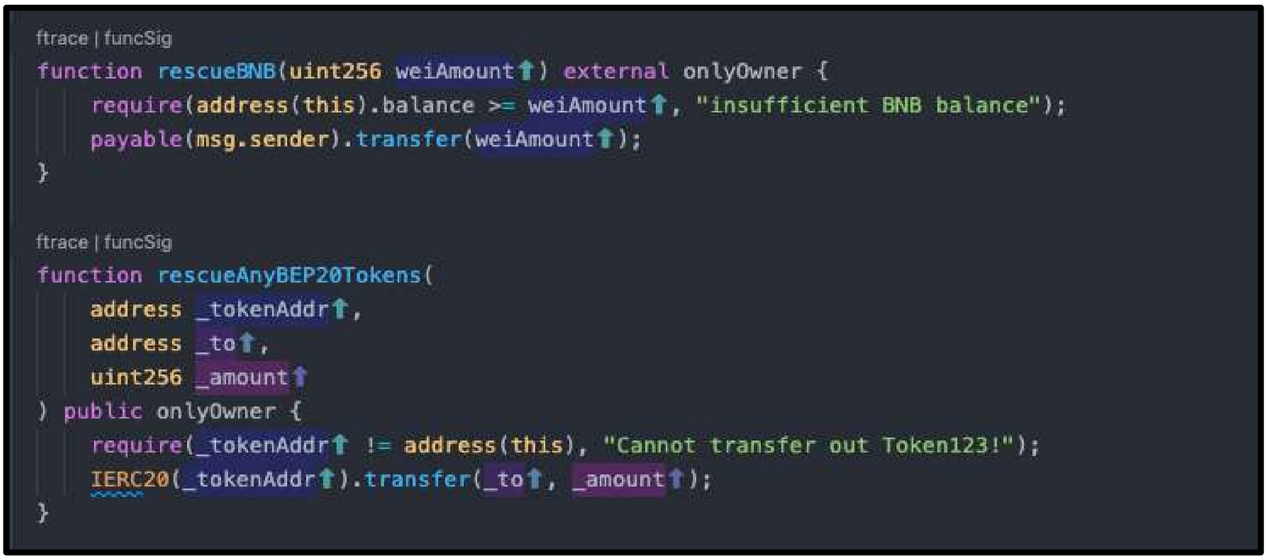
```
ftrace | funcSig
function updateSwapEnabled(bool _enabled) external onlyOwner {
    swapEnabled = _enabled;
}

ftrace | funcSig
```

- The owner can whitelist wallets from antibot.



```
ftrace | funcSig
function setAntibot(address account, bool state) external onlyOwner {
    require(_isBot[account] != state, "Value already set");
    _isBot[account] = state;
}
```

- The owner can get BNB and other tokens in contract to the owner wallet.



```
ftrace | funcSig
function rescueBNB(uint256 weiAmount) external onlyOwner {
    require(address(this).balance >= weiAmount, "insufficient BNB balance");
    payable(msg.sender).transfer(weiAmount);
}

ftrace | funcSig
function rescueAnyBEP20Tokens(
    address _tokenAddr,
    address _to,
    uint256 _amount
) public onlyOwner {
    require(_tokenAddr != address(this), "Cannot transfer out Token123!");
    IERC20(_tokenAddr).transfer(_to, _amount);
}
```

Audit conclusion

While conducting the audit of the Socaverse smart contract, it was observed that there is nothing alarming with the code.