KODABLE

Smart Contract SAFO

Contract: SAFO

Use this report to fix any issues and understand the code and follow recommendations.

SAFO Smart contract
Contract name:
SAFO-OTC yield Farm

Checklist

Compiler errors.		Passed
Possible delays in data delivery.		Passed
Timestamp dependence.		Passed
Integer Overflow and Underflow.		Passed
Race Conditions and Re-entrancy.		Passed
DoS with Revert.		Passed
DoS with block gas limit.		Passed
Methods execution permissions.		Passed
Economy model of the contract.		Passed
Private user data leaks.		Passed
Malicious Events Log.		Passed
Scoping and Declarations.		Passed
Uninitialized storage pointers.		Passed
Arithmetic accuracy.		Passed
Design Logic.		Passed
Impact of the exchange rate.		Passed
Oracle Calls.		Passed

Cross-function race conditions.		Passed
Fallback function security.		Passed
Safe Open Zeppelin contracts and implementation usage.		Passed

Implementations

This is a staking Contract

- User can claim the rewards. Rewards of user must be greater than zero.

Owner of this contract can withdraw the SAFO tokens from this contract.

Owner of this contract can withdraw the BNB amount from the contract.

Owner of this contract can withdraw the any bep20 token from this contract.

User can transfer the SAFO token to the contract.

Owner of this contract updates the SAFO bonus percentage value.

User can unstake from the pool.

User can pay in bnb and add the liquidity and purchase the SAFO.

Owner of this contract can update the acceptable slippage percentage value.

Owner of this contract can change the SAFO bonus active status to true/false.

User can redeem the SAFO tokens.

Owner of this contract withdraw the bnb from the contract exactly same as "beanfromsoldSAFO".

Owner of this contract can set the pool duration.

Owner of this contract can update the "reward rates" and "update at time" values.

Owner of this contract can update the team wallet address.

Owner of this contract can transfer the ownership of this contract.

Owner of this contract can set the early unstake fee percentage value.

Owner of this contract can set the referral percentage value.

Owner of this contract can set the referral limit value.

Owner of this contract set the reward rate, pool start time, pool end time and update at time.

Quick Stats:

Main Category	Subcategory	Result
Contract Programming	Solidity version not specified	Passed
	Solidity version too old	Passed
	Integer overflow/underflow	Passed
	Function input parameters lack of check	Passed
	Function input parameters check bypass	Passed
	Function access control lacks management	Passed
	Critical operation lacks event log	Passed
	Human/contract checks bypass	Passed
	Random number generation/use vulnerability	Passed
	Fallback function misuse	Passed
	Race condition	Passed
	Logical vulnerability	Passed
	Other programming issues	Passed
Code Specification	Visibility not explicitly declared	Passed
	Var. storage location not explicitly declared	Passed
	Use keywords/functions to be deprecated	Passed
	Other code specification issues	Passed
Gas Optimization	Assert () misuse	Passed
	High consumption ‘for/while’ loop	Passed
	High consumption ‘storage’ storage	Passed
	“Out of Gas” Attack	Passed
Business Risk	The maximum limit for mintage not set	Passed
	“Short Address” Attack	Passed
	“Double Spend” Attack	Passed

Summarised Audit Result: Medium risk

Executive Summary

According to the standard audit assessment, Customer`s solidity smart contract has medium level risk**.**

Kodable used critical analysis of the manual audit.
All issues found during analysis by Kodable manual review and applicable vulnerabilities are presented in the stat part.

Audit result: critical 0 - 3 medium - 0 low level problems.

Code standard

The SAFO Smart Contract protocol is made up of a single smart contract. Other inherited contracts include ReentrancyGuard. SAFO Smart Contract's libraries are part of its logical algorithm. They are smart contracts that include reusable code. Once deployed on the blockchain (once), it is assigned a unique address, and its properties and methods can be reused by other contracts in the protocol many times. The Kodable team has not provided scenario or unit test scripts, which would aid in the automated determination of code integrity.

The code is not commented in general. Commenting can provide extensive documentation for functions, return variables, and other elements.

Documentation

As mentioned above, it’s recommended to write comments in the smart contract code, so anyone can quickly understand the programming flow as well as complex code logic.

Use of Dependencies

According to our observations, libraries based on well-known industry standard open-source projects are used in this smart contract infrastructure. Even core code blocks are well-written and methodically. This smart contract has no interaction with other smart contracts.

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are typically easy to exploit
High	High-level vulnerabilities are difficult to exploit and also have significant impact on smart contract execution, e.g. public access
Medium	Medium-level vulnerabilities are important to fix but can't lose tokens
Low	Low-level vulnerabilities are mostly related to outdated, unused, and similar code snippets that have no significant impact on execution.
LoKodablest / Code Style / Best Practice	LoKodablest-level issues, code style etc and statements can’t affect smart contract execution and can be ignored.

Audit Findings

Critical

No Critical severity vulnerabilities were found.

High

No High severity vulnerabilities were found

Medium

1) Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.

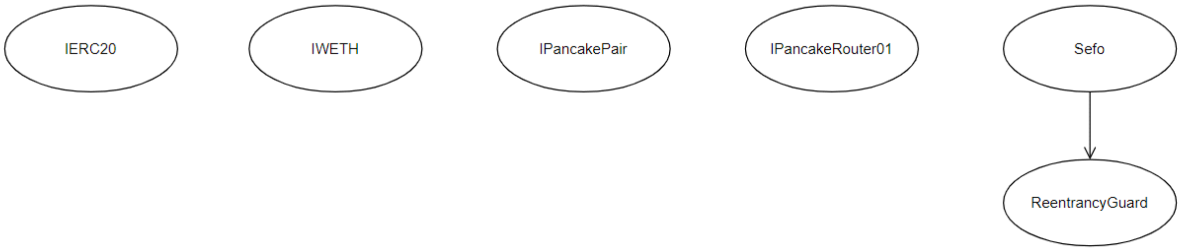
2) Use of "call": should be avoided whenever possible. It can lead to unexpected behavior if return value is not handled properly. Please use Direct Calls via specifying the called contract's interface.

3) The owner can change almost everything in the contract even after deployed it. The owner has the ability to modify the states of the contract.

Low

No Low severity vulnerabilities were found.

- Inheritance graph



Auto general report

- Files Description Table
-
- | File Name | SHA-1 Hash |
- |SAFO.sol|--------------|
- Contracts Description Table
- | 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❗ |
- ||||||
- | **IKodableTH** | Interface | |||
- | └ | deposit| External ❗ | 💵 |NO❗ |
- | └ | transfer| External ❗ | 🛑 |NO❗ |
- | └ | withdraw| External ❗ | 🛑 |NO❗ |
- ||||||
- | **IPancakePair** | Interface | |||
- | └ | totalSupply | External ❗ | |NO❗ |
- | └ | decimals | External ❗ | |NO❗ |
- | └ | symbol | External ❗ | |NO❗ |
- | └ | name | External ❗ | |NO❗ |
- | └ | balanceOf | External ❗ | |NO❗ |
- | └ | nonces| External ❗ | |NO❗ |
- | └ | PERMIT_TYPEHASH| External ❗ | |NO❗ |
- | └ | DOMAIN_SEPARATOR| External ❗ | |NO❗ |
- | └ | transfer | External ❗ | 🛑 |NO❗ |
- | └ | allowance | External ❗ | |NO❗ |
- | └ | approve | External ❗ | 🛑 |NO❗ |
- | └ | transferFrom | External ❗ | 🛑 |NO❗ |
- | └ | permit| External ❗ | 🛑 |NO❗ |
- | └ | MINIMUM_LIQUIDITY| External ❗ | |NO❗ |
- | └ | factory| External ❗ | |NO❗ |
- | └ | token0| External ❗ | |NO❗ |
- | └ | token1| External ❗ | |NO❗ |
- | └ | getReserves| External ❗ | |NO❗ |
- | └ | price0CumulativeLast| External ❗ | |NO❗ |
- | └ | price1CumulativeLast| External ❗ | |NO❗ |
- | └ | kLast| External ❗ | |NO❗ |
- | └ | mint| External ❗ | 🛑 |NO❗ |
- | └ | burn| External ❗ | 🛑 |NO❗ |
- | └ | swap | External ❗ | 🛑 |NO❗ |
- | └ | skim | External ❗ | 🛑 |NO❗ |
- | └ | sync| External ❗ | 🛑 |NO❗ |
- | └ | initialize| External ❗ | 🛑 |NO❗ |
- ||||||
- | **IPancakeRouter01** | Interface | |||
- | └ | quote| External ❗ | |NO❗ |
- | └ | getAmountIn| External ❗ | |NO❗ |
- | └ | getAmountOut| External ❗ | |NO❗ |

- | └ | getAmountsOut| External ❗ | |NO❗ |
- | └ | getAmountsIn| 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❗ |
- ||||||
- | **Presale** | Implementation | Ownable |||
- | └ | <Constructor> | ------- | 🛑 |NO❗ |
- | └ | <Receive Ether> | external ❗ | 💵 |NO❗ |
- | └ | purchaseSAFO | external ❗ | 💵 | NO❗ |
- | └ | redeemSAFO| External ❗ | 🛑 |NO❗ |
- | └ | fundSAFOs| External ❗ | 🛑 |NO❗ |
- | └ | defundSAFOs| External ❗ | 🛑 |onlyOwner|
- | └ | emergencyUnstake| External ❗ | 🛑 |NO❗ |
- | └ | claimRewards| External ❗ | 🛑 |NO❗ |
- | └ | setAcceptableSlippage| External ❗ | 🛑 | onlyOwner |
- | └ | setSAFOBonus| External ❗ | 🛑 | onlyOwner |
- | └ | setSAFOBonusActive| External ❗ | 🛑 | onlyOwner |
- | └ | withdrawBeansFromSoldSAFO| External ❗ | 🛑 | onlyOwner |
- | └ | setPoolDuration| External ❗ | 🛑 | onlyOwner |
- | └ | setPoolRewards| External ❗ | 🛑 | onlyOwner |
- | └ | topUpPoolRewards| External ❗ | 🛑 | onlyOwner |
- | └ | updateTeamWallet| External ❗ | 🛑 | onlyOwner |
- | └ | transferOwnership| External ❗ | 🛑 | onlyOwner |
- | └ | setEarlyUnstakeFee| External ❗ | 🛑 | onlyOwner |
- | └ | setRefferralPercentage| External ❗ | 🛑 | onlyOwner |
- | └ | setRefferralLimit| External ❗ | 🛑 | onlyOwner |
- | └ | emergencyRecoverBeans| External ❗ | 🛑 | onlyOwner |
- | └ | emergencyRecoverBEP20| External ❗ | 🛑 | onlyOwner |
-
- Legend
-
- | Symbol | Meaning |
- |:--------:|-----------|
- | 🛑 | Function can modify state |

- | 💵 | Function is payable |

Conclusion

Because the owner has a lot of rights and must take certain precautions, the smart contract code is medium risk. Several cautions were issued. Based on the provided objects, Kodable has run all conceivable tests. Kodable does not offer any such assurance of future results because there are an infinite number of conceivable test cases for such a complex smart contract protocol. In order to cover as many test cases as possible, Kodable scanned everything using the most recent static tools and manual observations. The report's stat section provided a high-level overview of the operation of Smart Contracts. The audit report lists all security flaws that were discovered.

The reviewed contract's security state is "Medium risk."

Suggested Solutions:

Kodable searches for quick fixes that live deployments can use, and then Kodable suggests specifications for remediation engineering in subsequent releases. After Kodable delivers our analysis and before the specifics are made public, the developers and deployment engineers should examine the mitigation and remediation recommendations. Successful mitigation and remediation is an ongoing collaborative process.