AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

KFC

DATED : 23 MAY 23'

Critical Risk

Issue: Ability to change the swap currency, affecting the swap and liquify mechanism

Severity: Critical
Function: setCurrency
Status: Not Resolved

Overview:
The owner can change the currency for swapping. This might affect the swap & liquify mechanism, potentially leading to unexpected behaviour or even reverted transactions.

Code:

```
function setCurrency(address _currency) public onlyOwner {
    currency = _currency;
    if (_currency == _swapRouter.WETH()) {
        currencyIsEth = true;
    } else {
        currencyIsEth = false;
    }
}
```

Suggestion:
Limit the owner's ability to change the swap currency or implement proper mechanisms to ensure the change doesn't affect the liquidity of the token adversely.

AUDIT SUMMARY

Project name – KFC

Date: 23 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	1	0	1	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

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/0x3CDed0726be6029fe2ab5d5906df5931C5eCe54F

Token Information

Name : KFC

Symbol : KFC

Decimals: 18

Network: BSC

Token Type:BEP20

Token Address:
0x96a48B8d5807909273D36ABAA06f7C8233bB58F3

Owner:
0xCbfBC639dd2704d5b1205E4b49091A075E5b7784
(at time of writing the audit)

Deployer:0x4a38Fe94CD63B705Cb6a1d5E6d707808a5019725

Token Information

Fees:

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 11%

Fees Privilige: Owner

Ownership :

0xCbfBC639dd2704d5b1205E4b49091A075E5b7784

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges:- - including in fees

- excluding from fees

- 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

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

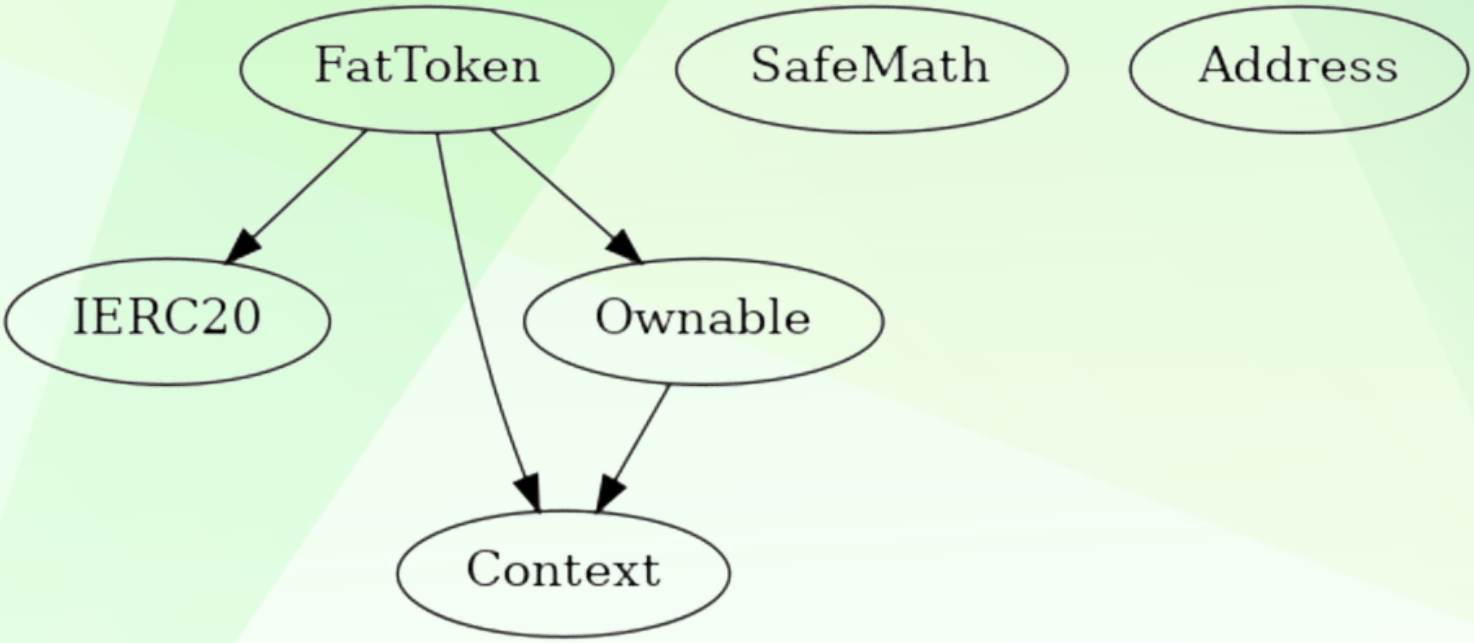
CLASSIFICATION OF RISK

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

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to change buy/sell/transfer taxes (static fees)
- 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 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 ❗
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO ❗
└	owner	Public ❗		NO ❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IUniswapV2Factory	Interface			
└	getPair	External ❗		NO ❗

CONTRACT ASSESMENT

└	createPair	External ❗	🛑	NO ❗
IUniswapV2Router01	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidity	External ❗	🛑	NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
TokenDistributor	Implementation			
└	<Constructor>	Public ❗	🛑	NO ❗
FatToken	Implementation	Context, IERC20, Ownable		
└	<Constructor>	Public ❗	🛑	NO ❗
└	setFundAddress	External ❗	🛑	onlyOwner
└	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 ❗
└	totalFees	Public ❗		NO ❗
└	deliver	Public ❗	🛑	NO ❗
└	reflectionFromToken	Public ❗		NO ❗
└	tokenFromReflection	Public ❗		NO ❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	_transferBothExcluded	Private 🔐	🛑	
└	setSwapPairList	External ❗	🛑	onlyOwner
└	multi_bclist	Public ❗	🛑	onlyOwner
└	setFeeWhiteList	External ❗	🛑	onlyOwner
└	setCurrency	Public ❗	🛑	onlyOwner
└	completeCustoms	External ❗	🛑	onlyOwner
└	setNumTokensSellToAddToLiquidity	Public ❗	🛑	onlyOwner
└	setSwapAndLiquifyEnabled	Public ❗	🛑	onlyOwner

CONTRACT ASSESMENT

└	<Receive Ether>	External ❗	💵	NO ❗
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	claimTokens	Public ❗	🛑	onlyOwner
└	calculateTaxFee_BUY	Private 🔐		
└	calculateTaxFee_SELL	Private 🔐		
└	calculateLiquidityFee_BUY	Private 🔐		
└	calculateLiquidityFee_SELL	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO ❗
└	_approve	Private 🔐	🛑	
└	isReward	Public ❗		NO ❗
└	setkb	Public ❗	🛑	onlyOwner
└	launch	External ❗	🛑	onlyOwner
└	disableSwapLimit	Public ❗	🛑	onlyOwner
└	disableWalletLimit	Public ❗	🛑	onlyOwner
└	disableChangeTax	Public ❗	🛑	onlyOwner
└	setCurrency	Public ❗	🛑	onlyOwner
└	changeSwapLimit	External ❗	🛑	onlyOwner
└	changeWalletLimit	External ❗	🛑	onlyOwner
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	swapTokensForEth_ETH	Private 🔐	🛑	
└	swapTokensForEth	Private 🔐	🛑	
└	addLiquidityETH	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	

CONTRACT ASSESMENT

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Variable FatToken._getValues(uint256,bool).rTransferAmount (contracts/Token.sol#1057) is too similar to FatToken._transferStandard(address,address,uint256).tTra
nsferAmount (contracts/Token.sol#1535)
Variable FatToken._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1578) is too similar to FatToken._transferBothExcluded(add
ress,address,uint256).tTransferAmount (contracts/Token.sol#946)
Variable FatToken._getValues(uint256,bool).rTransferAmount (contracts/Token.sol#1057) is too similar to FatToken._getValues(uint256,bool).tTransferAmount (contr
acts/Token.sol#1053)
Variable FatToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1533) is too similar to FatToken._transferBothExcluded(address
,address,uint256).tTransferAmount (contracts/Token.sol#946)
Variable FatToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#944) is too similar to FatToken._getTValues(uint256,bool).
tTransferAmount (contracts/Token.sol#1086)
Variable FatToken.reflectionFromToken(uint256,bool,bool).rTransferAmount (contracts/Token.sol#895) is too similar to FatToken._transferToExcluded(address,addres
s,uint256).tTransferAmount (contracts/Token.sol#1557)
Variable FatToken.reflectionFromToken(uint256,bool,bool).rTransferAmount (contracts/Token.sol#895) is too similar to FatToken._transferBothExcluded(address,addr
ess,uint256).tTransferAmount (contracts/Token.sol#946)
Variable FatToken._getValues(uint256,bool).rTransferAmount (contracts/Token.sol#1057) is too similar to FatToken._getTValues(uint256,bool).tTransferAmount (cont
racts/Token.sol#1086)
Variable FatToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#944) is too similar to FatToken._transferFromExcluded(addr
ess,address,uint256).tTransferAmount (contracts/Token.sol#1580)
Variable FatToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#944) is too similar to FatToken._transferStandard(address,
address,uint256).tTransferAmount (contracts/Token.sol#1535)
Variable FatToken._getValues(uint256,bool).rTransferAmount (contracts/Token.sol#1057) is too similar to FatToken._transferFromExcluded(address,address,uint256).
tTransferAmount (contracts/Token.sol#1580)
Variable FatToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1533) is too similar to FatToken._getTValues(uint256,bool).tTr
ansferAmount (contracts/Token.sol#1086)
Variable FatToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#944) is too similar to FatToken._transferToExcluded(addres
s,address,uint256).tTransferAmount (contracts/Token.sol#1557)
Variable FatToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#944) is too similar to FatToken._transferBothExcluded(addr
ess,address,uint256).tTransferAmount (contracts/Token.sol#946)
Variable FatToken._getValues(uint256,bool).rTransferAmount (contracts/Token.sol#1057) is too similar to FatToken._transferToExcluded(address,address,uint256).tT
ransferAmount (contracts/Token.sol#1557)
Variable FatToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1533) is too similar to FatToken._transferFromExcluded(address
,address,uint256).tTransferAmount (contracts/Token.sol#1580)
Variable FatToken._getValues(uint256,bool).rTransferAmount (contracts/Token.sol#1057) is too similar to FatToken._transferBothExcluded(address,address,uint256).
tTransferAmount (contracts/Token.sol#946)
Variable FatToken._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1099) is too similar to FatToken._transferToExcluded(address
,address,uint256).tTransferAmount (contracts/Token.sol#1557)
Variable FatToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#944) is too similar to FatToken._getValues(uint256,bool).t
TransferAmount (contracts/Token.sol#1053)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

FatToken.deadAddress (contracts/Token.sol#623) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

FatToken._tTotal (contracts/Token.sol#633) should be immutable
FatToken._tokenDistributor (contracts/Token.sol#630) should be immutable
FatToken.decimals (contracts/Token.sol#639) should be immutable
FatToken.enableKillBlock (contracts/Token.sol#644) should be immutable
FatToken.enableOffTrade (contracts/Token.sol#643) should be immutable
FatToken.enableRewardList (contracts/Token.sol#645) should be immutable
FatToken.name (contracts/Token.sol#637) should be immutable
FatToken.symbol (contracts/Token.sol#638) 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/0xd324710a7309fc8607c87bcb8d45d4d3b6f4d2a6d56b3dc2fc59a3594f3d523a

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x6690098aecce8f33a1655c5c70233e53fc02e6a2d63f6197276a531f342fc180

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xec49e0eec0f8e1d877c2b5af5f79496539ae602d37ef57ddea7e1149eab5b946

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xf3495d7ab1a5371f07fba9f168262f703ec22cbddeca6c400aa399d45ff2aead

5- Buying from a regular wallet (11% tax) (passed):
https://testnet.bscscan.com/tx/0x7f52081b6992d67efd171f25c09a3ad408b842954acae585a7b7ac5544cc0585

6- Selling from a regular wallet (10% tax) (passed):
https://testnet.bscscan.com/tx/0x2811c5652af5244e00eb33148f372f9d0c1932aed4f6e7217eb5ddb3e72dd7e8

7- Transferring from a regular wallet (11% tax) (passed):
https://testnet.bscscan.com/tx/0x4ab38ac900870544245 11c2d479cde3618aa9e04278be597754a16131ce21866

FUNCTIONAL TESTING

7- Internal swap (bnb fee and auto-liquidity)(passed):
https://testnet.bscscan.com/tx/0x4ab38ac9008705442451 1c2d479cde3618aa9e04278be597754a16131ce21866

FUNCTIONAL TESTING

Issue: Ability to change the swap currency, affecting the swap and liquify mechanism

Severity: Critical
Function: setCurrency
Status: Not Resolved

Overview:
The owner can change the currency for swapping. This might affect the swap & liquify mechanism, potentially leading to unexpected behaviour or even reverted transactions.

Code:

```
function setCurrency(address _currency) public onlyOwner {
    currency = _currency;
    if (_currency == _swapRouter.WETH()) {
        currencyIsEth = true;
    } else {
        currencyIsEth = false;
    }
}
```

Suggestion:
Limit the owner's ability to change the swap currency or implement proper mechanisms to ensure the change doesn't affect the liquidity of the token adversely.

FUNCTIONAL TESTING

Issue: Owner receives LP tokens from auto-liquidity, potentially enabling removal of a portion of the liquidity pool

Severity: Medium
Function: addLiquidityETH
Status: Not Resolved

Overview:
The owner receives LP tokens generated from auto-liquidity. Accumulated tokens can be used for removing a portion of the Liquidity pool.

Code:

```
function addLiquidityETH(uint256 tokenAmount, uint256 ethAmount) private {
  if (tokenAmount == 0) return;

  // approve token transfer to cover all possible scenarios
  _approve(address(this), address(_swapRouter), tokenAmount);

  // add the liquidity
  try
    _swapRouter.addLiquidityETH{value: ethAmount}(
      address(this),
      tokenAmount,
      0, // slippage is unavoidable
      0, // slippage is unavoidable
      address(fundAddress),
      block.timestamp
    )
  {} catch {
    emit failed_swap(2);
  }
}
```

Suggestion:
To maintain the stability of the liquidity pool, consider limiting the owner's ability to receive LP tokens from auto-liquidity, or provide a transparent mechanism for the use of such 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