auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

FitP

DATED : 9 July 23'

AUDIT SUMMARY

Project name – FitP

Date: 8 July, 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	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/0xacF8cFE42F7BFC2c60683A4c219Df45A45EaA284

Token Information

Token Name : FitPepe

Token Symbol: FitP

Decimals: 18

Token Supply: 1,000,000,000,000

Token Address:
0x4339d686cd4947cBE0c5257Ae0D916E96D1A4b7E

Checksum:
9a3da2962420e49aa3e157b7579246ebf53986ca

Owner:
0x567E9B6Ced720D40d5b1A771740542c7A3b347c8
(at time of writing the audit)

Deployer:
0x9A128da5E5a7E19be3875784FAF9656f96cebFC7

TOKEN OVERVIEW

Fees:

Buy Fees: 0-6%

Sell Fees: 0-8%

Transfer Fees: 0%

Fees Privilege: fees are immutable

Ownership: owned

Minting: none

Max Tx Amount/ Max Wallet Amount: Yes

Blacklist: No

Other Privileges: - Initial distribution of the tokens

- modifying fees

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

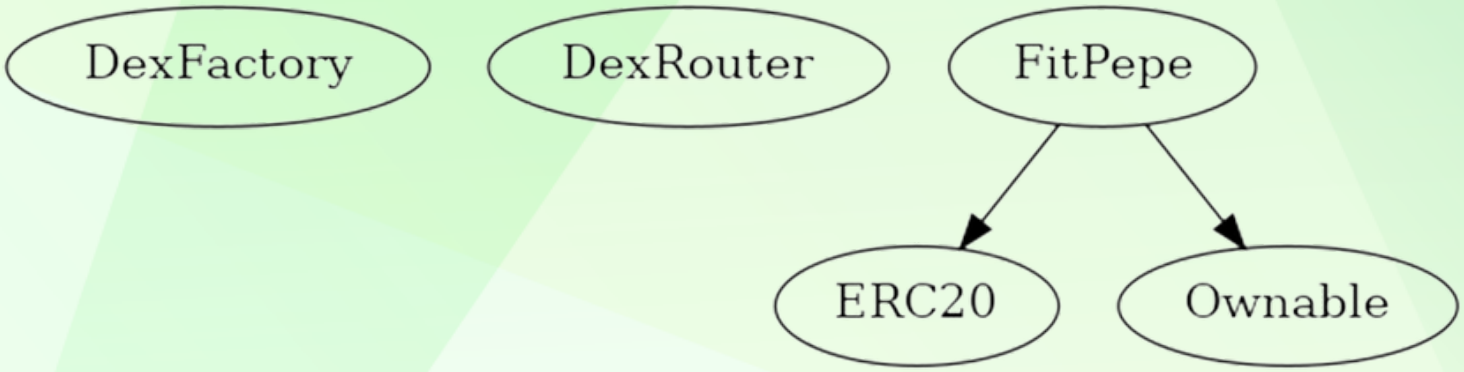
CLASSIFICATION OF RISK

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

INHERITANCE TREE



POINTS TO NOTE

- **Owner is able to change buy/sell fees within 0-6% for buys and 0-8% for sells**
- 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

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
DexFactory	Interface			
└	createPair	External ❗	🛑	NO ❗
DexRouter	Interface			
└	factory	External ❗		NO ❗
└	WETH	External ❗		NO ❗
└	addLiquidityETH	External ❗	💵	NO ❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO ❗
FitPepe	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🛑	ERC20
└	setMrketingWallet	External ❗	🛑	onlyOwner
└	setBuyBackWallet	External ❗	🛑	onlyOwner
└	setBuyTaxes	External ❗	🛑	onlyOwner
└	setSellTaxes	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	toggleSwapping	External ❗	🛑	onlyOwner
└	setWhitelistStatus	External ❗	🛑	onlyOwner
└	checkWhitelist	External ❗		NO ❗
└	_takeTax	Internal 🔒	🛑	
└	_transfer	Internal 🔒	🛑	
└	internalSwap	Internal 🔒	🛑	
└	swapToETH	Internal 🔒	🛑	
└	withdrawStuckETH	External ❗	🛑	onlyOwner
└	withdrawStuckTokens	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Context._msgData() (contracts/Token.sol#116-118) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#471-487) is never used and should be removed
SafeMath.add(uint256,uint256) (contracts/Token.sol#679-681) is never used and should be removed
SafeMath.div(uint256,uint256) (contracts/Token.sol#721-723) is never used and should be removed
SafeMath.div(uint256,uint256,string) (contracts/Token.sol#777-786) is never used and should be removed
SafeMath.mod(uint256,uint256) (contracts/Token.sol#737-739) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (contracts/Token.sol#803-812) is never used and should be removed
SafeMath.mul(uint256,uint256) (contracts/Token.sol#707-709) is never used and should be removed
SafeMath.sub(uint256,uint256) (contracts/Token.sol#693-695) is never used and should be removed
SafeMath.sub(uint256,uint256,string) (contracts/Token.sol#754-763) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (contracts/Token.sol#608-614) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (contracts/Token.sol#650-655) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (contracts/Token.sol#662-667) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (contracts/Token.sol#633-643) is never used and should be removed
SafeMath.trySub(uint256,uint256) (contracts/Token.sol#621-626) 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#9) 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#131) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#167) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#590) allows old versions
Pragma version^0.8.0 (contracts/Token.sol#825) allows old versions
Pragma version^0.8.17 (contracts/Token.sol#918) 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 FitPepe.internalSwap() (contracts/Token.sol#1083-1108):
        - (success) = marketingWallet.call{value: (totalMarketingFee * received) / totalShares}() (contracts/Token.sol#1101)
        - (success) = buybackWallet.call{value: address(this).balance}() (contracts/Token.sol#1106)
Low level call in FitPepe.withdrawStuckETH() (contracts/Token.sol#1121-1124):
        - (success) = address(msg.sender).call{value: address(this).balance}() (contracts/Token.sol#1122)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function DexRouter.WETH() (contracts/Token.sol#927) is not in mixedCase
Event FitPepemarketingWalletChanged(address) (contracts/Token.sol#978) is not in CapWords
Parameter FitPepe.setMrketingWallet(address)._newMarketing (contracts/Token.sol#999) is not in mixedCase
Parameter FitPepe.setBuyBackWallet(address)._newBuyBack (contracts/Token.sol#1004) is not in mixedCase
Parameter FitPepe.setBuyTaxes(uint256,uint256)._marketingTax (contracts/Token.sol#1009) is not in mixedCase
Parameter FitPepe.setBuyTaxes(uint256,uint256)._buybackTax (contracts/Token.sol#1009) is not in mixedCase
Parameter FitPepe.setSellTaxes(uint256,uint256)._marketingTax (contracts/Token.sol#1016) is not in mixedCase
Parameter FitPepe.setSellTaxes(uint256,uint256)._buybackTax (contracts/Token.sol#1016) is not in mixedCase
Parameter FitPepe.setSwapTokensAtAmount(uint256)._newAmount (contracts/Token.sol#1023) is not in mixedCase
Parameter FitPepe.setWhitelistStatus(address,bool)._wallet (contracts/Token.sol#1036) is not in mixedCase
Parameter FitPepe.setWhitelistStatus(address,bool)._status (contracts/Token.sol#1036) is not in mixedCase
Parameter FitPepe.checkWhitelist(address)._wallet (contracts/Token.sol#1041) is not in mixedCase
Parameter FitPepe.swapToETH(uint256)._amount (contracts/Token.sol#1110) is not in mixedCase
Parameter FitPepe.withdrawStuckTokens(address).BEP20_token (contracts/Token.sol#1126) is not in mixedCase
Constant FitPepe._totalSupply (contracts/Token.sol#953) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

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/0xc04d7bcbb844027f2ae33e0e0daa37fa3cbaa0758e03f8241652e2f4a55fb66b

2- Buying when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x1be4338df258e9d012adbebe330af1871e7aaddefff5f55a47303e49a9118a28

3- Selling when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xcba608e6f5383fff21b824814d747c2119ba2231c919b4d4bb513a69e9707cb2

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0xef7100eb5a634b3a911a646c73e351b214ca905136b273e121682f9ca872264f

5- Buying(0-6% tax) (passed):
https://testnet.bscscan.com/tx/0x7da8f36a1054490cb3a754d3b48855e6246b995f723469cf2ca379ad31375c86

6- Selling (0-6% tax) (passed):
https://testnet.bscscan.com/tx/0x9a640ae39d86c52300827c27ae86481dd4c9e4ae30d35ec117030f9ae96ee4f3

FUNCTIONAL TESTING

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

https://testnet.bscscan.com/tx/0x953cef0c7cce92ddc9298bdca6e3840691096e9691a964f2983016a733b80be3

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

https://testnet.bscscan.com/tx/0x9a640ae39d86c52300827c27ae86481dd4c9e4ae30d35ec117030f9ae96ee4f3

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