



Smart Contract Audit

FOR

Wojak Original

DATED : 08 May 23'



AUDIT SUMMARY

Project name – Wojak Original

Date: 08 May, 2023

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

Audit Status: **Failed**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	2	0	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 done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

<https://testnet.bscscan.com/token/0x3D54B2786c91abf02f97A7c1b12bD20bc06C5Df7>



Token Information

Token Name : Wojak Original

Token Symbol: Wojak

Decimals: 5

Token Supply:69,000,000,000

Token Address:

0xC8Bad1E2A992e585e6c7f31649B6e95140682f9a

Checksum:

05f70e518e3aab63f00e899780eaeff23a977e55

Owner:

0xC544b1291817f8733B5067e7f3F1F56c0Dc67B27

Deployer:

0xC544b1291817f8733B5067e7f3F1F56c0Dc67B27



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

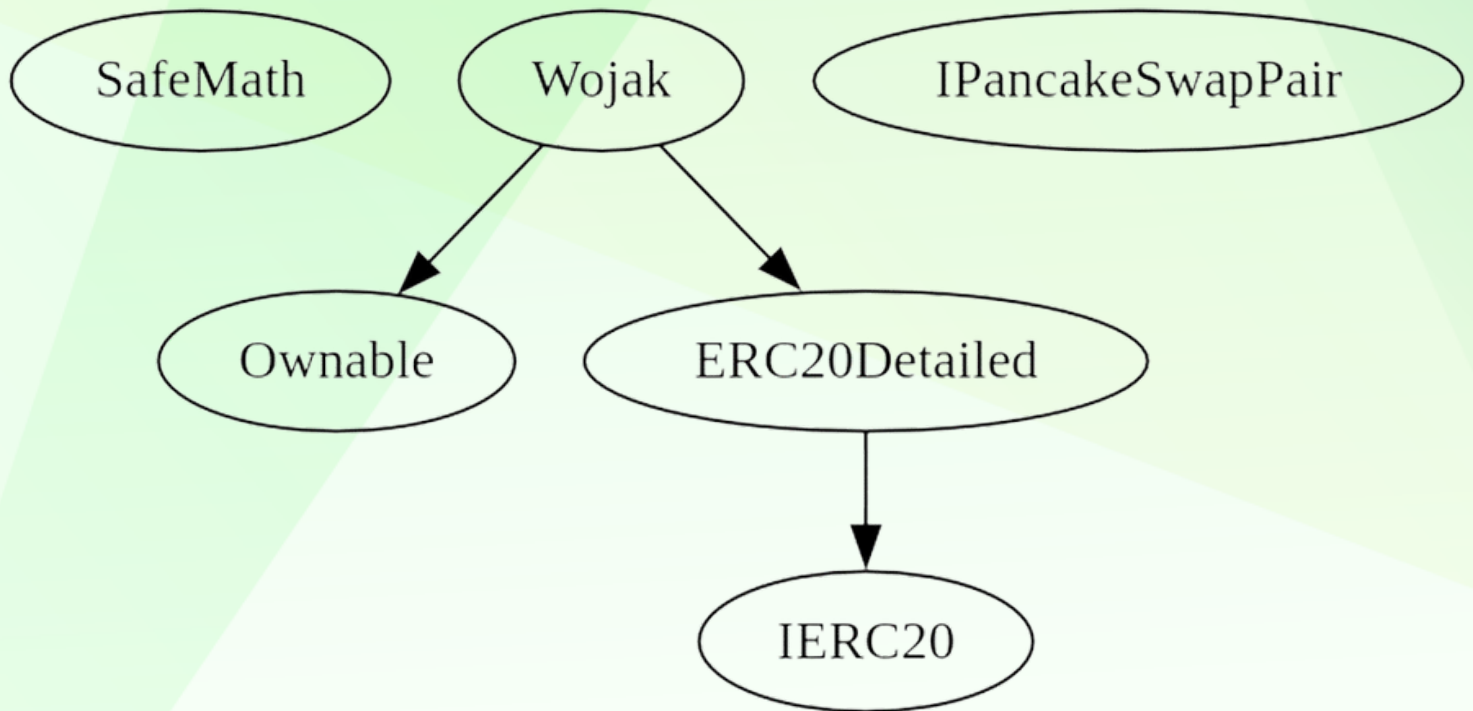
Findings

Severity

Found

◆ Critical	2
◆ High-Risk	0
◆ Medium-Risk	1
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is not able to modify buy/sell fees (3% for each)
 - Owner is not able to set transfer fee (0% always)
 - **Owner is able to blacklist an arbitrary address.**
 - **Owner is able to disable trades**
 - Owner is not able to set max buy/sell/transfer/hold amount
 - Owner is not able to mint new tokens
 - Token supply and holders balance increases 0.83% each 8 hours. This also leads to a decrease in price since pool balance will be increased (rebase)
-

TOKEN DISTRIBUTION

It should be noted that the owner currently holds 100% of the total supply. However, information about the distribution of these tokens is not available, and it is recommended that investors exercise caution when considering this aspect.

CONTRACT ASSESMENT

Contract	Type	Bases			
:-----: :-----: :-----: :-----: :-----:					
L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
SafeMathInt Library					
L	mul	Internal	🔒		
L	div	Internal	🔒		
L	sub	Internal	🔒		
L	add	Internal	🔒		
L	abs	Internal	🔒		
SafeMath Library					
L	add	Internal	🔒		
L	sub	Internal	🔒		
L	sub	Internal	🔒		
L	mul	Internal	🔒		
L	div	Internal	🔒		
L	div	Internal	🔒		
L	mod	Internal	🔒		
IERC20 Interface					
L	totalSupply	External	⚠️		NO!
L	balanceOf	External	⚠️		NO!
L	allowance	External	⚠️		NO!
L	transfer	External	⚠️	🛑	NO!
L	approve	External	⚠️	🛑	NO!
L	transferFrom	External	⚠️	🛑	NO!
IPancakeSwapPair Interface					
L	name	External	⚠️		NO!
L	symbol	External	⚠️		NO!
L	decimals	External	⚠️		NO!
L	totalSupply	External	⚠️		NO!
L	balanceOf	External	⚠️		NO!
L	allowance	External	⚠️		NO!
L	approve	External	⚠️	🛑	NO!
L	transfer	External	⚠️	🛑	NO!
L	transferFrom	External	⚠️	🛑	NO!
L	DOMAIN_SEPARATOR	External	⚠️		NO!
L	PERMIT_TYPEHASH	External	⚠️		NO!
L	nonces	External	⚠️		NO!
L	permit	External	⚠️	🛑	NO!
L	MINIMUM_LIQUIDITY	External	⚠️		NO!




CONTRACT ASSESMENT

```
|  | 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! |
|  |  |  |  |  |
| **IPancakeSwapRouter** | 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! |
|  | removeLiquidityETHSupportingFeeOnTransferTokens | External ! |  | NO! |
|  | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! |  | NO! |
|  | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! |  | NO! |
|  | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! |  | NO! |
|  | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! |  | NO! |
|  |  |  |  |  |
| **IPancakeSwapFactory** | Interface | | |
```



CONTRACT ASSESMENT

```
|  | feeTo | External ! | | NO! |
|  | feeToSetter | External ! | | NO! |
|  | getPair | External ! | | NO! |
|  | allPairs | External ! | | NO! |
|  | allPairsLength | External ! | | NO! |
|  | createPair | External ! | | NO! |
|  | setFeeTo | External ! | | NO! |
|  | setFeeToSetter | External ! | | NO! |
|  |  |  |  |  |
| **Ownable** | Implementation | | |
|  | <Constructor> | Public ! | | NO! |
|  | owner | Public ! | | NO! |
|  | isOwner | Public ! | | NO! |
|  | renounceOwnership | Public ! | | onlyOwner |
|  | transferOwnership | Public ! | | onlyOwner |
|  | _transferOwnership | Internal | | |
|  |  |  |  |  |
| **ERC20Detailed** | Implementation | IERC20 | | |
|  | <Constructor> | Public ! | | NO! |
|  | name | Public ! | | NO! |
|  | symbol | Public ! | | NO! |
|  | decimals | Public ! | | NO! |
|  |  |  |  |  |
| **Wojak** | Implementation | ERC20Detailed, Ownable | | |
|  | <Constructor> | Public ! | | ERC20Detailed Ownable |
|  | rebase | Internal | | |
|  | transfer | External ! | | validRecipient |
|  | transferFrom | External ! | | validRecipient |
|  | _basicTransfer | Internal | | |
|  | _transferFrom | Internal | | |
|  | takeFee | Internal | | |
|  | addLiquidity | Internal | | swapping |
|  | shouldTakeFee | Internal | | |
|  | shouldRebase | Internal | | |
|  | shouldAddLiquidity | Internal | | |
|  | setAutoRebase | External ! | | onlyOwner |
|  | setFeeFlag | External ! | | onlyOwner |
|  | setAutoAddLiquidity | External ! | | onlyOwner |
|  | allowance | External ! | | NO! |
|  | decreaseAllowance | External ! | | NO! |
|  | increaseAllowance | External ! | | NO! |
```

CONTRACT ASSESMENT

^L	approve	External !		NO!
^L	checkFeeExempt	External !		NO!
^L	getCirculatingSupply	Public !		NO!
^L	isNotInSwap	External !		NO!
^L	manualSync	External !		NO!
^L	setFeeReceivers	External !		onlyOwner
^L	getLiquidityBacking	Public !		NO!
^L	setWhitelist	External !		onlyOwner
^L	setBotBlacklist	External !		onlyOwner
^L	setPairAddress	Public !		onlyOwner
^L	setLP	External !		onlyOwner
^L	totalSupply	External !		NO!
^L	balanceOf	External !		NO!
^L	isContract	Internal 		
^L	<Receive Ether>	External !		NO!

Legend

Symbol	Meaning
	Function can modify state
	Function is payable



STATIC ANALYSIS

```
Function IPancakeSwapPair.PERMIT_TYPEHASH() (contracts/Token.sol#161) is not in mixedCase
Function IPancakeSwapPair.MINIMUM_LIQUIDITY() (contracts/Token.sol#192) is not in mixedCase
Function IPancakeSwapRouter.WETH() (contracts/Token.sol#232) is not in mixedCase
Parameter Wojak.setAutoRebase(bool). _flag (contracts/Token.sol#789) is not in mixedCase
Parameter Wojak.setFeeFlag(bool). _flag (contracts/Token.sol#798) is not in mixedCase
Parameter Wojak.setAutoAddLiquidity(bool). _flag (contracts/Token.sol#802) is not in mixedCase
Parameter Wojak.checkFeeExempt(address). _addr (contracts/Token.sol#862) is not in mixedCase
Parameter Wojak.setFeeReceivers(address,address). _autoLiquidityReceiver (contracts/Token.sol#882) is not in mixedCase
Parameter Wojak.setFeeReceivers(address,address). _firePit (contracts/Token.sol#883) is not in mixedCase
Parameter Wojak.setWhitelist(address). _addr (contracts/Token.sol#897) is not in mixedCase
Parameter Wojak.setBotBlacklist(address,bool). _botAddress (contracts/Token.sol#902) is not in mixedCase
Parameter Wojak.setBotBlacklist(address,bool). _flag (contracts/Token.sol#903) is not in mixedCase
Parameter Wojak.setPairAddress(address). _pairAddress (contracts/Token.sol#912) is not in mixedCase
Parameter Wojak.setLP(address). _address (contracts/Token.sol#916) is not in mixedCase
Variable Wojak._name (contracts/Token.sol#527) is not in mixedCase
Variable Wojak._symbol (contracts/Token.sol#528) is not in mixedCase
Variable Wojak._decimals (contracts/Token.sol#529) is not in mixedCase
Variable Wojak._isFeeExempt (contracts/Token.sol#532) is not in mixedCase
Variable Wojak.DEAD (contracts/Token.sol#551) is not in mixedCase
Variable Wojak.ZERO (contracts/Token.sol#552) is not in mixedCase
Variable Wojak._autoRebase (contracts/Token.sol#573) is not in mixedCase
Variable Wojak._feeFlag (contracts/Token.sol#574) is not in mixedCase
Variable Wojak._autoAddLiquidity (contracts/Token.sol#575) is not in mixedCase
Variable Wojak._initRebaseStartTime (contracts/Token.sol#576) is not in mixedCase
Variable Wojak._lastRebasedTime (contracts/Token.sol#577) is not in mixedCase
Variable Wojak._lastAddLiquidityTime (contracts/Token.sol#578) is not in mixedCase
Variable Wojak._totalSupply (contracts/Token.sol#579) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IPancakeSwapRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#237) is too similar to IPancakeSwapRouter.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#238)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

SafeMathInt.MAX_INT256 (contracts/Token.sol#13) is never used in SafeMathInt (contracts/Token.sol#11-44)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Wojak.DEAD (contracts/Token.sol#551) should be constant
Wojak.ZERO (contracts/Token.sol#552) should be constant
Wojak._decimals (contracts/Token.sol#529) should be constant
Wojak._name (contracts/Token.sol#527) should be constant
Wojak._symbol (contracts/Token.sol#528) should be constant
Wojak.feeDenominator (contracts/Token.sol#549) should be constant
Wojak.firePitFee (contracts/Token.sol#547) should be constant
Wojak.liquidityFee (contracts/Token.sol#546) should be constant
Wojak.swapEnabled (contracts/Token.sol#558) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

Wojak._initRebaseStartTime (contracts/Token.sol#576) should be immutable
Wojak.pair (contracts/Token.sol#560) should be immutable
Wojak.router (contracts/Token.sol#559) should be immutable
Wojak.totalFee (contracts/Token.sol#548) should be immutable
Wojak.treasuryReceiver (contracts/Token.sol#555) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

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

All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):

<https://testnet.bscscan.com/tx/0x93cf64196bc1aba7c78b81b4f329a6d4ff84453a8af2bf6af9d8a0fdca9011f2>

2- Buying when excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x27fd0e7c2591564c90651973c2d4fe4188d00740c50d290c2cbc7711390d48e5>

3- Selling when excluded (0% tax) (passed):

<https://testnet.bscscan.com/tx/0x47389ccba452735c8099a4a3534acda37411e0489b885ebf13e340ada8f13c65>

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

<https://testnet.bscscan.com/tx/0x8d8ec85c4d9a1164620e48b08a7a2d92f345b4a599b02a81508197c9aeb78724>

5- Buying when not excluded (3% tax) (passed):

<https://testnet.bscscan.com/tx/0x48b7142f1f2927481bf8cab9ec8df1134378d31daf6d13249422b5f4819c6cf5>

6- Selling when not excluded (3% tax) (passed):

<https://testnet.bscscan.com/tx/0x76dc573aa0c4bf53ffd69e0484c331fdb389142e9e88333d0204e24e19e33e0d>



FUNCTIONAL TESTING

7- Transferring when not excluded (3% tax) (passed):

<https://testnet.bscscan.com/tx/0x7342eb8d5f14676ad326b79f72e12aad2286d591f3519345326886091e9abec8>

MANUAL TESTING

Centralization - Owner can blacklist wallets

Severity: Critical

Function: setBotBlacklist

Status: Not Resolved

Overview:

The owner is able to blacklist an arbitrary address . Blacklisted addresses are not able to buy/sell/transfer tokens.

Blacklisting liquidity pool means that trades will be completely disabled even for owner

```
function setBotBlacklist(
    address _botAddress,
    bool _flag
) external onlyOwner {
    require(
        isContract(_botAddress),
        "only contract address, not allowed exteranlly owned account"
    );
    blacklist[_botAddress] = _flag;
}
```

Recommendation:

to address this issue there are multiple options

- Renounce the ownership
- Implement an auto-blacklist functionality which blacklists bad actors in a reasonable and limited time after launch

MANUAL TESTING

Logical – pair contract can be changed

Severity: Critical

Function: setLP

Status: not resolved

Overview:

pairContract can be changed to a new address, if new address is not a contract or is a contract that does not have “sync” method or has a malicious “sync” method, rebase function will be reverting the whole transaction

```
function setLP(address _address) external onlyOwner {
    pairContract = IPancakeSwapPair(_address);
}

function rebase() internal {
    if (inSwap) return;
    uint256 rebaseRate;
    uint256 deltaTime = block.timestamp - _lastRebasedTime;
    uint256 times = deltaTime.div(8 hours);
    uint256 epoch = times.mul(480);

    rebaseRate = 8300;

    for (uint256 i = 0; i < times; i++) {
        _totalSupply = _totalSupply
            .mul((10 ** RATE_DECIMALS).add(rebaseRate))
            .div(10 ** RATE_DECIMALS);
    }

    _gonsPerFragment = TOTAL_GONS.div(_totalSupply);
    _lastRebasedTime = _lastRebasedTime.add(times.mul(8 hours));

    pairContract.sync();

    emit LogRebase(epoch, _totalSupply);
}
```

Recommendation:

Ensure that pairContract can not be changed later

MANUAL TESTING

Logical – Outdated compiler version

Severity: Medium

Status: not resolved

Overview:

Compiler version is 0.7.14, this compiler version is outdated and doesn't support internal handling of overflow/underflows

Recommendation:

Change the compiler version to $\geq 0.8.0$



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 specialize 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>
