AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Body AI

DATED : 19 MAR 23'

AUDIT SUMMARY

Project name – Body AI

Date: 19 March, 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 Testnet network:

all tests were done on Bsc Testnet 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/0x79cEE3A8E16 3dbd1B6172EbC80b9d73d22473C8C

Token Information

Token Name : Body AI

Token Symbol: BAIT

Decimals: 18

Token Supply: 1,000,000,000

Token Address:
0x859aC2100b85868bf7387D27e339da502FD1Eabc

Checksum:
6f4fe1a9273198cf26370ff2b4f644bf19b72f7a

Owner:
0x5b6C5eaDf3BA926f928DE834Ccd6562F4a10d34c
(at time of writing the audit)

TOKEN OVERVIEW

Fees:

Buy Fees: 3%

Sell Fees: 3%

Transfer Fees: 0%

Fees Privilige: none

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: none

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

◆ **Critical**

◆ **High-Risk**

◆ **Medium-Risk**

◆ **Low-Risk**

◆ **Gas Optimization /Suggestion**

Description

These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.

A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

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
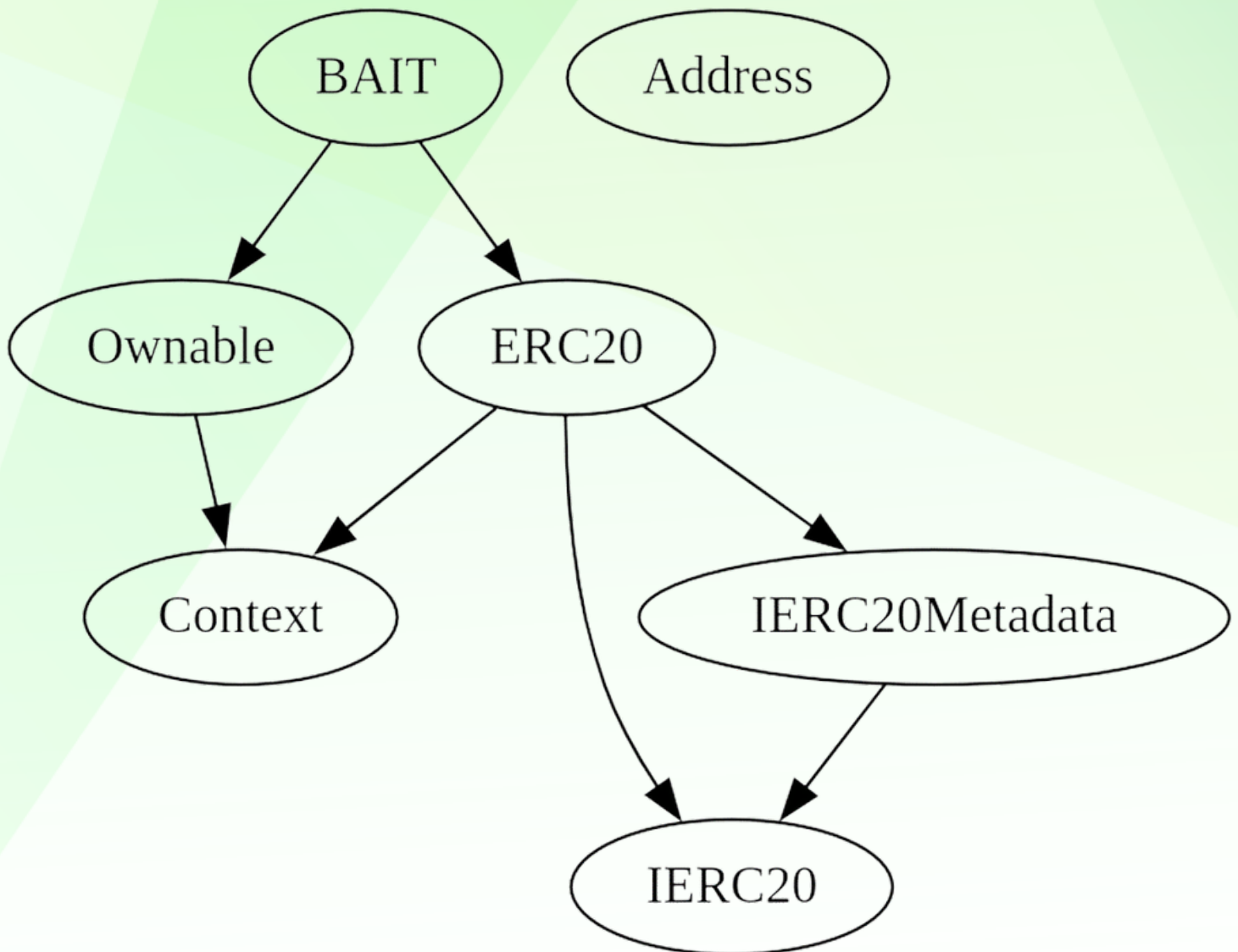
POINTS TO NOTE

- Owner is not able to change buy/sell/transfer fees (3% buy, 3% sell, 0% transfer)
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens
- Owner must enable trading in order for investors to be able to buy/sell/transfer
- Owner of the contract is safu dev, hence enabling trades is guaranteed

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
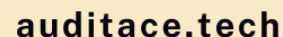
Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
IUniswapV2Pair	Interface			
└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	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❗

CONTRACT ASSESMENT

IUniswapV2Router01	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❗
IUniswapV2Router02	Interface	IUniswapV2Router01			
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑		NO❗
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑		NO❗
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑		NO❗
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵		NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑		NO❗
IERC20	Interface				
└	totalSupply	External ❗			NO❗
└	balanceOf	External ❗			NO❗
└	transfer	External ❗	🛑		NO❗
└	allowance	External ❗			NO❗
└	approve	External ❗	🛑		NO❗
└	transferFrom	External ❗	🛑		NO❗
IERC20Metadata	Interface	IERC20			
└	name	External ❗			NO❗
└	symbol	External ❗			NO❗
└	decimals	External ❗			NO❗

CONTRACT ASSESMENT

Address	Library				
└	isContract	Internal 🔒			
└	sendValue	Internal 🔒	🛑		
└	functionCall	Internal 🔒	🛑		
└	functionCall	Internal 🔒	🛑		
└	functionCallWithValue	Internal 🔒	🛑		
└	functionCallWithValue	Internal 🔒	🛑		
└	functionStaticCall	Internal 🔒			
└	functionStaticCall	Internal 🔒			
└	functionDelegateCall	Internal 🔒	🛑		
└	functionDelegateCall	Internal 🔒	🛑		
└	verifyCallResultFromTarget	Internal 🔒			
└	verifyCallResult	Internal 🔒			
└	_revert	Private 🔐			
Context	Implementation				
└	_msgSender	Internal 🔒			
└	_msgData	Internal 🔒			
Ownable	Implementation	Context			
└	<Constructor>	Public ❗	🛑		NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
ERC20	Implementation	Context, IERC20, IERC20Metadata			
└	<Constructor>	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❗
└	_transfer	Internal 🔒	🛑		
└	_mint	Internal 🔒	🛑		
└	_burn	Internal 🔒	🛑		

CONTRACT ASSESMENT

└	_approve	Internal 🔒	🛑		
└	_beforeTokenTransfer	Internal 🔒	🛑		
└	_afterTokenTransfer	Internal 🔒	🛑		
BAIT	Implementation	ERC20, Ownable		
└	<Constructor>	Public ❗	🛑	ERC20
└	<Receive Ether>	External ❗	💵	NO❗
└	claimStuckTokens	External ❗	🛑	onlyOwner
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isExcludedFromFees	Public ❗		NO❗
└	enableTrading	External ❗	🛑	onlyOwner
└	_transfer	Internal 🔒	🛑	
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	swapAndSendMarketing	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Address._revert(bytes,string) (contracts/Token.sol#549-564) is never used and should be removed
Address.functionCall(address,bytes) (contracts/Token.sol#408-419) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/Token.sol#421-427) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/Token.sol#429-441) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#443-463) is never used and should be removed
Address.functionDelegateCall(address,bytes) (contracts/Token.sol#492-502) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#504-517) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/Token.sol#465-475) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#477-490) is never used and should be removed
Address.isContract(address) (contracts/Token.sol#391-393) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/Token.sol#537-547) is never used and should be removed
Address.verifyCallResultFromTarget(address,bool,bytes,string) (contracts/Token.sol#519-535) is never used and should be removed
Context._msgData() (contracts/Token.sol#572-575) is never used and should be removed
ERC20._burn(address,uint256) (contracts/Token.sol#762-777) 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#23) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.19 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#395-406):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#401)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#443-463):
        - (success,returndata) = target.call{value: value}(data) (contracts/Token.sol#453-455)
Low level call in Address.functionStaticCall(address,bytes,string) (contracts/Token.sol#477-490):
        - (success,returndata) = target.staticcall(data) (contracts/Token.sol#482)
Low level call in Address.functionDelegateCall(address,bytes,string) (contracts/Token.sol#504-517):
        - (success,returndata) = target.delegatecall(data) (contracts/Token.sol#509)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#85) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#87) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#118) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#158) is not in mixedCase
Parameter BAIT.setSwapEnabled(bool)._enabled (contracts/Token.sol#979) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#573)" inContext (contracts/Token.sol#567-576)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (contracts/Token.sol#163) is too similar to IUniswapV2Router01.addLiquidit
y(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#164)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

BAIT.creator (contracts/Token.sol#812) should be immutable
BAIT.developmentWallet (contracts/Token.sol#819) should be immutable
BAIT.marketingFeeOnBuy (contracts/Token.sol#814) should be immutable
BAIT.marketingFeeOnSell (contracts/Token.sol#815) should be immutable
BAIT.marketingWallet (contracts/Token.sol#817) should be immutable
BAIT.treasuryWallet (contracts/Token.sol#818) should be immutable
BAIT.uniswapV2Pair (contracts/Token.sol#808) should be immutable
BAIT.uniswapV2Router (contracts/Token.sol#807) 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 issues found

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding Liquidity (Passed):
liquidity added on Pancakeswap V2:

https://testnet.bscscan.com/tx/0x00c93464e4e566fb23dfa54b6c
aceb6663af639dc4dda0ba44b536f0d99c5da0

2- Buying when excluded (0%)(Passed):

https://testnet.bscscan.com/tx/0x0aec111ecd3218af5dbc9529af35
083a8c4ec2c7c62d53f2e253b5cfa793aeae

3- Selling when excluded (0%)(Passed):

https://testnet.bscscan.com/tx/0x13cfee1f7e2c6f6950cc87f79dd0
860bf1a924d57877f61f3d519623d3e7d18b

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

https://testnet.bscscan.com/tx/0x4d14b1b27fe8ef4d0d3796c3584
f96f24e368987a0cc7bf533128e1cee18a2be

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

https://testnet.bscscan.com/tx/0x92ef6f24a8116bd963c2e55d6b8
7a87769e58232ec74d08f471142d3615b7e1e

FUNCTIONAL TESTING

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

https://testnet.bscscan.com/tx/0x36135041c5fe91805d20b509eb7ec411adf07f904b0b80cd46615da655b8107d

7- Transferring when not excluded(0% tax) (passed):

https://testnet.bscscan.com/tx/0x3178a13c8ad3bb846a63c9ecba73c46d334a7570afa3007ba836144557fcc5f4

8- Internal swap (passed):

fee wallets received BNB with same amount (1/3 of contract BNB)

https://testnet.bscscan.com/address/0x55d2279d69c729284a1e3d01ca70676a58fa63d3#internaltx
https://testnet.bscscan.com/address/0x956cf7601835e3a2e238dcab94c2d5c40e1deced#internaltx
https://testnet.bscscan.com/address/0xefa0991489caeabefcc13cdc6d168d4b348dd4ba#internaltx

MANUAL TESTING

No Issues Found

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