AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Optimus AI

DATED : 25 MAR 23'

AUDIT SUMMARY

Project name – Optimus AI

Date: 25 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	2	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	2	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/0xADaD0f09593F2F7AC6A29703b7a8fAaC7aB6D95b

Token Information

Token Name : Optimus AI

Token Symbol: Optimus AI

Decimals: 9

Token Supply: 1,000,000,000

Token Address:
0xAd3063FE9dF7355fC6E008c04f8Da6e02B40304E

Checksum:
956f6e7d5feaabd3fb70b30e61815ac048ceb1c8

Owner:
0xAE6A8763191534AD4cf5FbaAEF53b1788148B501
(at time of writing the audit)

TOKEN OVERVIEW

Fees:

Buy Fees: 10%

Sell Fees: 10%

Transfer Fees: 10%

Fees Privilige: Owner

Ownership : Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: exlcuding from rewards - including in rewards - changing swap threshold

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

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

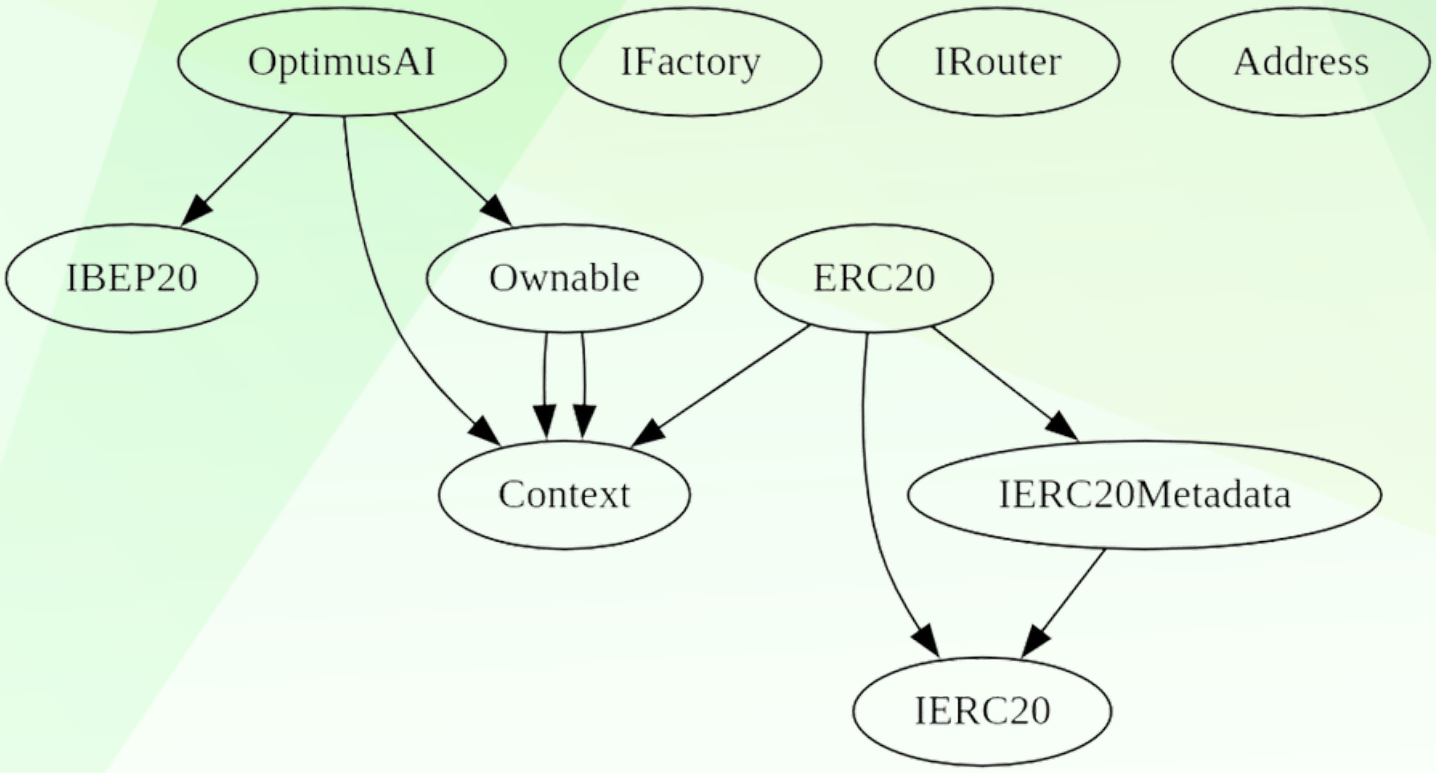
CLASSIFICATION OF RISK

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	2 (Resolved)
◆ Medium-Risk	0
◆ Low-Risk	0
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to set buy/sell/transfer fee each one up to 10%
- Owner must enable trading for investors to be able to trade
- 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

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**
IBEP20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
IFactory	Interface			
└	createPair	External ❗	🛑	NO❗
IRouter	Interface			
└	factory	External ❗		NO❗
└	WETH	External ❗		NO❗
└	addLiquidityETH	External ❗	💵	NO❗
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑	NO❗
Address	Library			
└	sendValue	Internal 🔒	🛑	
OptimusAI	Implementation	Context, IBEP20, Ownable		
└	<Constructor>	Public ❗	🛑	NO❗
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗

CONTRACT ASSESMENT

└	allowance	Public ❗		|NO❗
└	approve	Public ❗	🛑	|NO❗
└	transferFrom	Public ❗	🛑	|NO❗
└	increaseAllowance	Public ❗	🛑	|NO❗
└	decreaseAllowance	Public ❗	🛑	|NO❗
└	transfer	Public ❗	🛑	|NO❗
└	isExcludedFromReward	Public ❗		|NO❗
└	reflectionFromToken	Public ❗		|NO❗
└	EnableTrading	External ❗	🛑	onlyOwner
└	updatedeadline	External ❗	🛑	onlyOwner
└	tokenFromReflection	Public ❗		|NO❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	excludeFromFee	Public ❗	🛑	onlyOwner
└	includeInFee	Public ❗	🛑	onlyOwner
└	isExcludedFromFee	Public ❗		|NO❗
└	_reflectRfi	Private 🔐	🛑		
└	_takeLiquidity	Private 🔐	🛑		
└	_takeMarketing	Private 🔐	🛑		
└	_takeOps	Private 🔐	🛑		
└	_takeDev	Private 🔐	🛑		
└	_getValues	Private 🔐			
└	_getTValues	Private 🔐			
└	_getRValues1	Private 🔐			
└	_getRValues2	Private 🔐			
└	_getRate	Private 🔐			
└	_getCurrentSupply	Private 🔐			
└	_approve	Private 🔐	🛑		
└	_transfer	Private 🔐	🛑		
└	_tokenTransfer	Private 🔐	🛑		
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	addLiquidity	Private 🔐	🛑		
└	swapTokensForBNB	Private 🔐	🛑		
└	bulkExcludeFee	External ❗	🛑	onlyOwner
└	updateMarketingWallet	External ❗	🛑	onlyOwner
└	updateDevWallet	External ❗	🛑	onlyOwner
└	setTaxes	Public ❗	🛑	onlyOwner
└	setSellTaxes	Public ❗	🛑	onlyOwner
└	updateOpsWallet	External ❗	🛑	onlyOwner
└	updateSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	updateSwapEnabled	External ❗	🛑	onlyOwner

CONTRACT ASSESMENT

└	rescueBNB	External ❗	🛑	onlyOwner
└	rescueAnyBEP20Tokens	Public ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO❗
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 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_spendAllowance	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	Internal 🔒	🛑	
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❗
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		

CONTRACT ASSESMENT

Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

OptimusAI.includeInReward(address) (contracts/Token.sol#417-428) has costly operations inside a loop:
        - _excluded.pop() (contracts/Token.sol#424)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#59-62) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

OptimusAI._rTotal (contracts/Token.sol#178) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#20) 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#139-150):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#145)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#115) is not in mixedCase
Struct OptimusAI.valuesFromGetValues (contracts/Token.sol#215-229) is not in CapWords
Function OptimusAI.EnableTrading() (contracts/Token.sol#383-388) is not in mixedCase
Parameter OptimusAI.updatedeadline(uint256)._deadline (contracts/Token.sol#390) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256,uint256)._rfi (contracts/Token.sol#794) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256,uint256)._marketing (contracts/Token.sol#795) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256,uint256)._ops (contracts/Token.sol#796) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256,uint256)._liquidity (contracts/Token.sol#797) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256,uint256)._dev (contracts/Token.sol#798) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256,uint256)._rfi (contracts/Token.sol#809) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256,uint256)._marketing (contracts/Token.sol#810) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256,uint256)._ops (contracts/Token.sol#811) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256,uint256)._liquidity (contracts/Token.sol#812) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256,uint256)._dev (contracts/Token.sol#813) is not in mixedCase
Parameter OptimusAI.updateSwapEnabled(bool)._enabled (contracts/Token.sol#836) is not in mixedCase
Parameter OptimusAI.rescueAnyBEP20Tokens(address,address,uint256)._tokenAddr (contracts/Token.sol#848) is not in mixedCase
Parameter OptimusAI.rescueAnyBEP20Tokens(address,address,uint256)._to (contracts/Token.sol#849) is not in mixedCase
Parameter OptimusAI.rescueAnyBEP20Tokens(address,address,uint256)._amount (contracts/Token.sol#850) is not in mixedCase
Constant OptimusAI._decimals (contracts/Token.sol#174) is not in UPPER_CASE_WITH_UNDERSCORES
Variable OptimusAI.genesis_block (contracts/Token.sol#182) is not in mixedCase
Constant OptimusAI._name (contracts/Token.sol#190) is not in UPPER_CASE_WITH_UNDERSCORES
Constant OptimusAI._symbol (contracts/Token.sol#191) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#60)" inContext (contracts/Token.sol#54-63)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

OptimusAI._lastSell (contracts/Token.sol#169) is never used in OptimusAI (contracts/Token.sol#153-860)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

OptimusAI._tTotal (contracts/Token.sol#177) should be constant
OptimusAI.deadWallet (contracts/Token.sol#185) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

OptimusAI.pair (contracts/Token.sol#172) should be immutable
OptimusAI.router (contracts/Token.sol#171) 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/0x0037388763bc1854cd6ee750e376ab5016da45618c99a8781b1be8f8f630cc6d

2- Buying when trading not enabled (owner%)(Passed):

https://testnet.bscscan.com/tx/0x928e1bcffc8755d5e3dab4dad7e9688ecc49d00ffcd18518509fa91a9633c95e

3- Selling when trading not enabled (0%)(Passed):

https://testnet.bscscan.com/tx/0xb17b6858b4db57459acec3e2369e5f6dcc3a5d6b3baf8a158965e35d338205ff

4- Transferring when trading not enabled (0% tax) (passed):

https://testnet.bscscan.com/tx/0xffcd137adc9f8a102abdb10e641b77bef88368872c983399edf5b6594fd6a3ec

5- Buying when trading enabled (up to 10% tax) (passed):

https://testnet.bscscan.com/tx/0xbea668394662de0c551a9a8bef4d5ef78e1443e47ab1d0e2a80d6849cfffddae

FUNCTIONAL TESTING

6- Selling when trading enabled (up to 10% tax) (passed):

https://testnet.bscscan.com/tx/0x1d380cf2a3536c0c9aa6a640b1305f19ec0537c94c418eae1b6079a005d27488

7- Transferring when trading enabled (up to 0% tax) (passed):

https://testnet.bscscan.com/tx/0x9732c11f41ced3afd8795932398414c0a851f7d006fee10a2cbf56e611178200

8- Internal swap (passed):
All fee wallets received BNB
https://testnet.bscscan.com/tx/0x9732c11f41ced3afd8795932398414c0a851f7d006fee10a2cbf56e611178200

9- Auto Liquidity (passed):
https://testnet.bscscan.com/token/0x86410ea66bee49143ab915ab7e3fcf86ffc36a74?a=0x000000000000000000000000000000000000dead

MANUAL TESTING

Centralization - Owner must enable trading

Severity: High
Function: EnableTrading
Lines: 328
Status: Resolved
Overview:
The owner must activate trading for investors to buy, sell, or transfer tokens. If trading remains disabled, token holders will be unable to trade their tokens.

```
function EnableTrading() external onlyOwner {
    require(!tradingEnabled, "Cannot re-enable trading");
    tradingEnabled = true;
    swapEnabled = true;
    genesis_block = block.number;
}
```

Recommendation:
Incorporate a safety mechanism that allows investors to activate trading if a specified duration has elapsed since the conclusion of the presale.
Since contract is owned by safu dev, enabling trades is guaranteed.

MANUAL TESTING

Logical – Setting swap threshold to 0

Severity: High
Function: updateSwapTokensAtAmount
Lines: 761
Status: not resolved
Overview:
setting swap threshold to 0 can disable sells if contract balance is more than threshold.

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner {
    require(
        amount <= 1e7,
        "Cannot set swap threshold amount higher than 1% of tokens"
    );
    swapTokensAtAmount = amount * 10 ** _decimals;
}
```

Recommendation:
ensure that swap threshold can not be zero.
Since contract is owned by safu dev, swap threshold will not be set to 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 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