



# 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 Privilege:** Owner

---

**Ownership :** Owned

---

**Minting:** No mint function

---

**Max Tx Amount/ Max Wallet Amount:** No

---

**Blacklist:** No

---

**Other Privileges:** excluding 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 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 Limit and 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



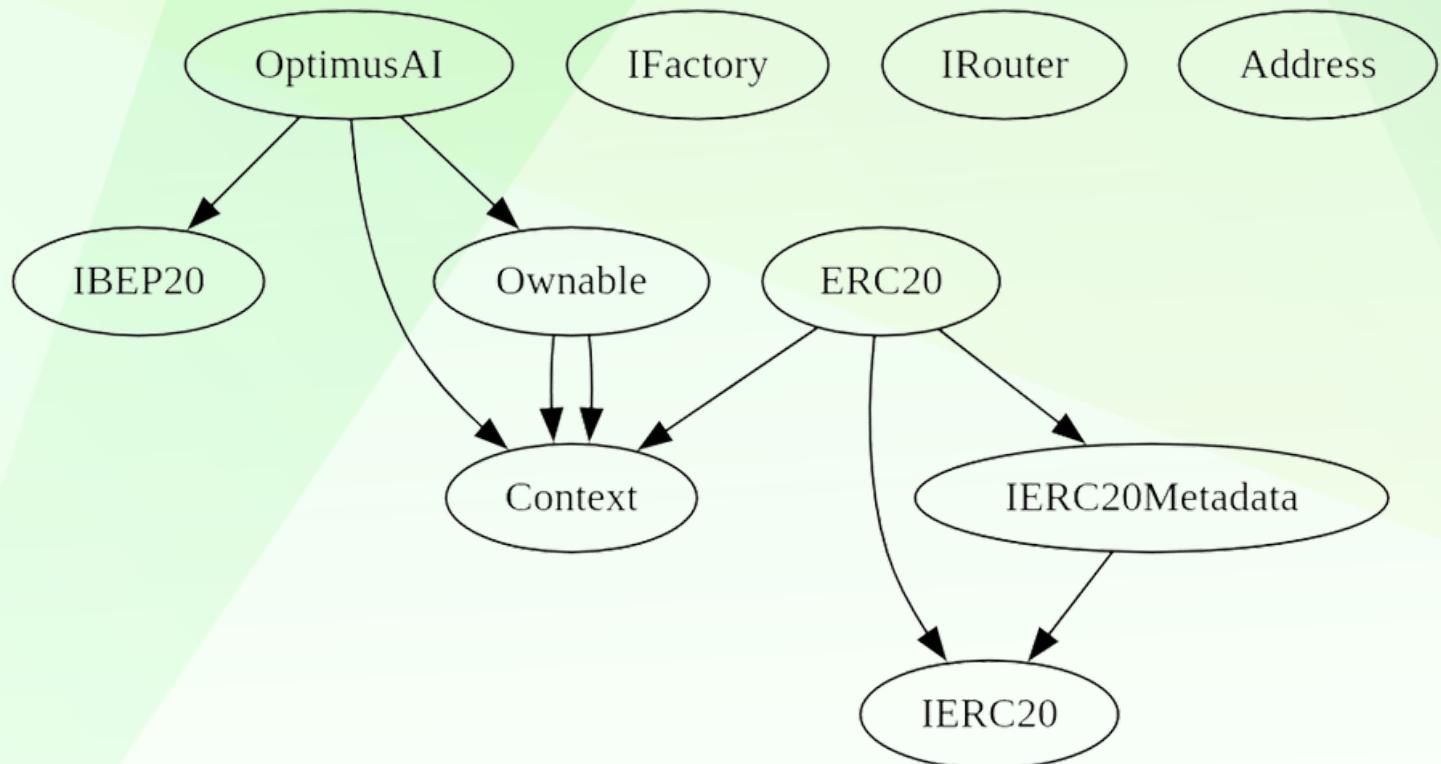
# CLASSIFICATION OF RISK

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	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 ASSESSMENT

Contract	Type	Bases			
----- ----- ----- ----- -----					
**Function Name**   **Visibility**   **Mutability**   **Modifiers**					
**IBEP20**   Interface					
L   totalSupply   External !     NO!					
L   balanceOf   External !     NO!					
L   transfer   External !     NO!					
L   allowance   External !     NO!					
L   approve   External !     NO!					
L   transferFrom   External !     NO!					
**Context**   Implementation					
L   _msgSender   Internal 🔒					
L   _msgData   Internal 🔒					
**Ownable**   Implementation   Context					
L   <Constructor>   Public !     NO!					
L   owner   Public !     NO!					
L   renounceOwnership   Public !     NO!					
L   transferOwnership   Public !     NO!					
L   _setOwner   Private 🗂️					
**IFactory**   Interface					
L   createPair   External !     NO!					
**IRouter**   Interface					
L   factory   External !     NO!					
L   WETH   External !     NO!					
L   addLiquidityETH   External !     NO!					
L   swapExactTokensForETHSupportingFeeOnTransferTokens   External !     NO!					
**Address**   Library					
L   sendValue   Internal 🔒					
**OptimusAI**   Implementation   Context, IBEP20, Ownable					
L   <Constructor>   Public !     NO!					
L   name   Public !     NO!					
L   symbol   Public !     NO!					
L   decimals   Public !     NO!					
L   totalSupply   Public !     NO!					
L   balanceOf   Public !     NO!					



# CONTRACT ASSESSMENT

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

# CONTRACT ASSESSMENT

L	rescueBNB	External !		onlyOwner	
L	rescueAnyBEP20Tokens	Public !		onlyOwner	
L	<Receive Ether>	External !		NO!	
\*\*ERC20\*\*	Implementation	Context, IERC20, IERC20Metadata			
L	<Constructor>	Public !		NO!	
L	name	Public !		NO!	
L	symbol	Public !		NO!	
L	decimals	Public !		NO!	
L	totalSupply	Public !		NO!	
L	balanceOf	Public !		NO!	
L	transfer	Public !		NO!	
L	allowance	Public !		NO!	
L	approve	Public !		NO!	
L	transferFrom	Public !		NO!	
L	increaseAllowance	Public !		NO!	
L	decreaseAllowance	Public !		NO!	
L	\_transfer	Internal 			
L	\_mint	Internal 			
L	\_burn	Internal 			
L	\_approve	Internal 			
L	\_spendAllowance	Internal 			
L	\_beforeTokenTransfer	Internal 			
L	\_afterTokenTransfer	Internal 			
\*\*IERC20\*\*	Interface				
L	totalSupply	External !		NO!	
L	balanceOf	External !		NO!	
L	transfer	External !		NO!	
L	allowance	External !		NO!	
L	approve	External !		NO!	
L	transferFrom	External !		NO!	
\*\*IERC20Metadata\*\*	Interface	IERC20			
L	name	External !		NO!	
L	symbol	External !		NO!	
L	decimals	External !		NO!	
\*\*Context\*\*	Implementation				
L	\_msgSender	Internal 			
L	\_msgData	Internal 			



# CONTRACT ASSESSMENT

	**Ownable**	Implementation   Context	
L	<Constructor>	Public !	①   NO !
L	owner	Public !	NO !
L	_checkOwner	Internal 🔒	
L	renounceOwnership	Public !	①   onlyOwner
L	transferOwnership	Public !	①   onlyOwner
L	_transferOwnership	Internal 🔒	①
	**SafeMath**	Library	
L	tryAdd	Internal 🔒	
L	trySub	Internal 🔒	
L	tryMul	Internal 🔒	
L	tryDiv	Internal 🔒	
L	tryMod	Internal 🔒	
L	add	Internal 🔒	
L	sub	Internal 🔒	
L	mul	Internal 🔒	
L	div	Internal 🔒	
L	mod	Internal 🔒	
L	sub	Internal 🔒	
L	div	Internal 🔒	
L	mod	Internal 🔒	

## Legend

	Symbol	Meaning	
-----: -----			
	①	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.updatedDeadline(uint256).deadline (contracts/Token.sol#390) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256).rfi (contracts/Token.sol#794) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256).marketing (contracts/Token.sol#795) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256).ops (contracts/Token.sol#796) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256).liquidity (contracts/Token.sol#797) is not in mixedCase
Parameter OptimusAI.setTaxes(uint256,uint256,uint256,uint256).dev (contracts/Token.sol#798) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256).rfi (contracts/Token.sol#809) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256).marketing (contracts/Token.sol#810) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256).ops (contracts/Token.sol#811) is not in mixedCase
Parameter OptimusAI.setSellTaxes(uint256,uint256,uint256,uint256).liquidity (contracts/Token.sol#812) is not in mixedCase
Parameter OptimusAI.setSellTaxes(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/0xbea668394662de0c551a9a8bef4d5ef78e1443e47ab1d0e2a80d6849cfffdde>

---



# 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/0x9732c11f41ced3af8795932398414c0a851f7d006fee10a2cbf56e611178200>

## 8- Internal swap (**passed**):

All fee wallets received BNB

<https://testnet.bscscan.com/tx/0x9732c11f41ced3af8795932398414c0a851f7d006fee10a2chf56e611178200>

## 9- Auto Liquidity (passed):

<https://testnet.bscscan.com/token/0x86410ea66bee49143ab915ab7e3fcf86ffc36a742>



# 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



# Social Media Overview

---

**Here are the Social Media Accounts of  
Optimus AI**



**[https://t.me/OptimusAI\\_Community](https://t.me/OptimusAI_Community)**



**[https://twitter.com/Optimus\\_AI\\_BSC](https://twitter.com/Optimus_AI_BSC)**



**[https://t.me/OptimusAI\\_Channel](https://t.me/OptimusAI_Channel)**

---



# 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://t.me/Audit_Ace)**



**[https://twitter.com/auditace\\_](https://twitter.com/auditace_)**



**<https://github.com/Audit-Ace>**

---