AuditAce
FROM INCEPTION TO SUCCESS

# Smart Contract Audit

FOR

## QuantumAI

**DATED : 23 FEB 23'**

# AUDIT SUMMARY

**Project name** – QuantumAI

**Date**: 23 February, 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 | 3 |
| 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. You can check this tests in "functionality tests" section of the report.

### 3- Slither :The code has undergone static analysis using Slither.

# TESTNET LINKS

---

**All tests were done using this contract,BSC Testnet**

https://testnet.bscscan.com/token/0xB8c81a5557446d040BcD98B98E1560AF12Bcc61F

**Token Address: :**
0x4949dA31bF62B764A31dd23f9A4D9Fa72eBb4eD8

**Checksum:**
7f4637c4c0993af2f6a3874de24406b78e17d63c560bf5b3e6f76b0b4edc8fe0

**Deployer:**
0x329456727c84315a0A06Ac79545162c4179507C9

**Owner**:
0x329456727c84315a0A06Ac79545162c4179507C9

**Network:** Arbitrum

**THIS TOKEN IS ON ARBITRUM NETWORK**

# 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**
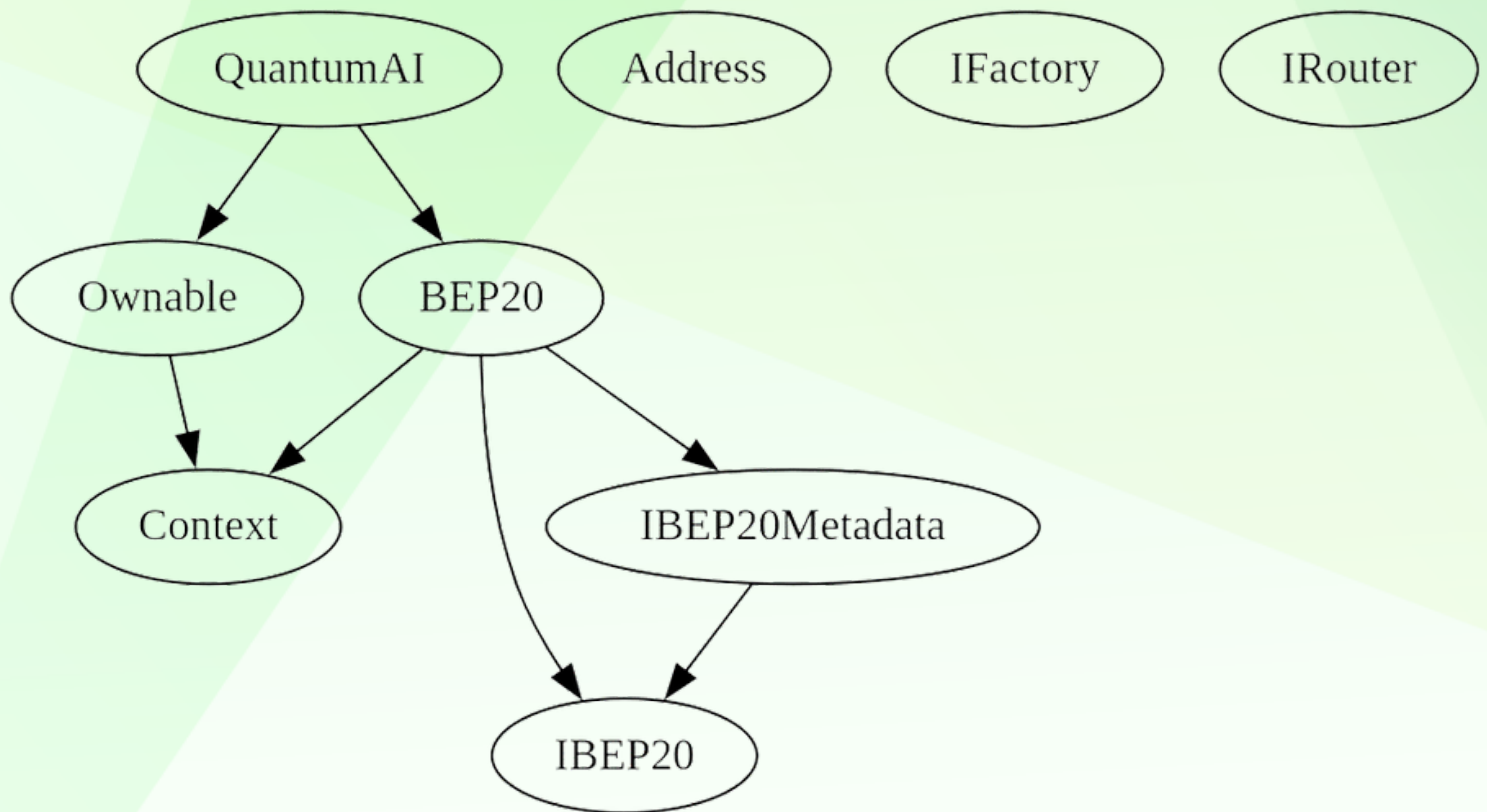
0

◆ **Gas Optimization / Suggestions**

3

# INHERITANCE TREE

# POINTS TO NOTE

- **Up to 5 blocks after launch will have 99% tax**

- **Owner is not able to set buy/sell/transfer taxes (0% static)**

- **Owner is not able to blacklist an arbitrary wallet**

- **Owner is not able to set max buy/sell/transfer amounts**

- **Owner is not able to disable trades**

- **Owner is not able to mint new tokens**

# CONTRACT ASSESMENT

| Contract | Type | Bases | | |
|:----------:|:------------------:|:----------------:|:----------------:|:---------------:|
| └ | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |
| **Context** | Implementation | | | |
| └ | _msgSender | Internal 🔒 | | |
| └ | _msgData | Internal 🔒 | | |
| | | | | |
| **IBEP20** | Interface | | | |
| └ | totalSupply | External ❗ | | NO❗ |
| └ | balanceOf | External ❗ | | NO❗ |
| └ | transfer | External ❗ | 🛑 | NO❗ |
| └ | allowance | External ❗ | | NO❗ |
| └ | approve | External ❗ | 🛑 | NO❗ |
| └ | transferFrom | External ❗ | 🛑 | NO❗ |
| | | | | |
| **IBEP20Metadata** | Interface | IBEP20 | | |
| └ | name | External ❗ | | NO❗ |
| └ | symbol | External ❗ | | NO❗ |
| └ | decimals | External ❗ | | NO❗ |
| | | | | |
| **BEP20** | Implementation | Context, IBEP20, IBEP20Metadata | | |
| └ | <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 🔒 | 🛑 | |
| └ | _tokengeneration | Internal 🔒 | 🛑 | |
| └ | _approve | Internal 🔒 | 🛑 | |
| | | | | |
| **Address** | Library | | | |
| └ | sendValue | Internal 🔒 | 🛑 | |
| | | | | |
| **Ownable** | Implementation | Context | | |

# CONTRACT ASSESMENT

| └ | <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❗ |
| | | | | | |
| **QuantumAI** | Implementation | BEP20, Ownable | | |
| └ | <Constructor> | Public ❗ | 🛑 | BEP20 |
| └ | approve | Public ❗ | 🛑 |NO❗ |
| └ | transferFrom | Public ❗ | 🛑 |NO❗ |
| └ | increaseAllowance | Public ❗ | 🛑 |NO❗ |
| └ | decreaseAllowance | Public ❗ | 🛑 |NO❗ |
| └ | transfer | Public ❗ | 🛑 |NO❗ |
| └ | _transfer | Internal 🔒 | 🛑 | |
| └ | Liquify | Private 🔒 | 🛑 | lockTheSwap |
| └ | swapTokensForETH | Private 🔒 | 🛑 | |
| └ | addLiquidity | Private 🔒 | 🛑 | |
| └ | updateLiquidityProvide | External ❗ | 🛑 | onlyOwner |
| └ | updateLiquidityTreshhold | External ❗ | 🛑 | onlyOwner |
| └ | EnableTrading | External ❗ | 🛑 | onlyOwner |
| └ | updatedeadline | External ❗ | 🛑 | onlyOwner |
| └ | updateMarketingWallet | External ❗ | 🛑 | onlyOwner |
| └ | updateExemptFee | External ❗ | 🛑 | onlyOwner |
| └ | bulkExemptFee | External ❗ | 🛑 | onlyOwner |
| └ | rescueBNB | External ❗ | 🛑 | onlyOwner |
| └ | rescueBSC20 | External ❗ | 🛑 | onlyOwner |
| └ | <Receive Ether> | External ❗ | 💵 |NO❗ |

| Symbol | Meaning |
|:--------:|-----------|
| 🛑 | Function can modify state |
| 💵 | Function is payable |

# STATIC ANALYSIS

**A static analysis of contract's source code has been performed using slither. No major issues were found in the output**

```
super._transfer(sender,address(this),feeAmount) (contracts/Token.sol#550)
Reentrancy in QuantumAI.transferFrom(address,address,uint256) (contracts/Token.sol#494-509):
        External calls:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#499)
                - router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (contracts/Token.sol#665-672)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#344)
                - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#651-657)
                - address(marketingWallet).sendValue(marketingAmt) (contracts/Token.sol#637)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (contracts/Token.sol#499)
                - router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,deadWallet,block.timestamp) (contracts/Token.sol#665-672)
                - (success) = recipient.call{value: amount}() (contracts/Token.sol#344)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (contracts/Token.sol#333)
                - _approve(sender,_msgSender(),currentAllowance - amount) (contracts/Token.sol#506)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Context._msgData() (contracts/Token.sol#14-17) 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#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 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#338-349):
        - (success) = recipient.call{value: amount}() (contracts/Token.sol#344)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Variable BEP20._balances (contracts/Token.sol#70) is not in mixedCase
Variable BEP20._allowances (contracts/Token.sol#72) is not in mixedCase
Function IRouter.WETH() (contracts/Token.sol#402) is not in mixedCase
Function QuantumAI.Liquify(uint256,QuantumAI.Taxes) (contracts/Token.sol#601-640) is not in mixedCase
Parameter QuantumAI.updateLiquidityTreshhold(uint256).new_amount (contracts/Token.sol#679) is not in mixedCase
Function QuantumAI.EnableTrading() (contracts/Token.sol#687-692) is not in mixedCase
Parameter QuantumAI.updatedeadline(uint256)._deadline (contracts/Token.sol#694) is not in mixedCase
Parameter QuantumAI.updateExemptFee(address,bool)._address (contracts/Token.sol#705) is not in mixedCase
Variable QuantumAI.genesis_block (contracts/Token.sol#437) is not in mixedCase
Constant QuantumAI.deadWallet (contracts/Token.sol#442-443) 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#15)" inContext (contracts/Token.sol#9-18)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

QuantumAI._lastSell (contracts/Token.sol#456) is never used in QuantumAI (contracts/Token.sol#425-732)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

QuantumAI.launchtax (contracts/Token.sol#439) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

QuantumAI.pair (contracts/Token.sol#429) should be immutable
QuantumAI.router (contracts/Token.sol#428) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

# FUNCTIONAL TESTING

**Functionality Tests
Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1**

**increased fees to max limit**

**1- Adding liquidity** **(passed):**
https://testnet.bscscan.com/tx/0x5299c79c8701eb3a4d4484ac27
93bdbe90bd7433cdc176de927ab2890348f68e

**2- Buying (0% )** **(passed):**
https://testnet.bscscan.com/tx/0x0562cc4fccf5f378a2d4a1170c81
c72a21dfc6eb208b5909802d9cfab1d89af1

**3- Selling (0% )** **(passed):**
https://testnet.bscscan.com/tx/0x9c0f1ad90c7ef96f96b51904a10f
827fce1d56c447b2b3584e70b6dc9d6950d5

**4- Transferring (0% )** **(passed):**
https://testnet.bscscan.com/tx/0x52cc259c649da91a59ed0ea37c
4f6bca72a1a241e756e86a894e7a3f500d5545

# MANUAL TESTING

## Suggestions & Gas optimizations:

### Suggestions:

- Since contract can not have fees, its suggested to remove unused functions and logic

### Gas Optimizations:

- Since contract can not have fees, then some functions and code can increase gas usage unnecessary
- Consider optimizing the contract's code to reduce its overall gas usage.

# 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**