QuiDax Smart Contract Preliminary Audit Report

Project Synopsis

Project Name	**Quidax**
Platform	Ethereum, Solidity
Github Repo	Not Provided
Deployed Contract	Not Deployed
Total Duration	2 Days
Timeline of Audit	**2nd May 2021 to 4th May 2021**

Contract Details

Total Contract(s)	1
Name of Contract(s)	Quidax.sol
Language	Solidity
Commit Hash	**Null**

Contract Vulnerabilities Synopsis

Issues	Open Issues	Closed Issues
Critical Severity	0	0
Medium Severity	0	0
Low Severity	4	0
Information	1	0
Total Found	5	0

Detailed Results

The contract has gone through several stages of the audit procedure that includes structural analysis, automated testing, manual code review etc.

All the issues have been explained and discussed in detail below. Along with the explanation of the issue found during the audit, the recommended way to overcome the issue or improve the code quality has also been mentioned.

A. Contract Name: QuiDax

High Severity Issues
None Found

Medium Severity Issues
None Found

Low Severity Issues
A.1 Comparison to boolean Constant
Line no: 419,428, 437, 448, 605, 606, 607, 627, 628, 650, 651, 692, 693, 694,

Description:

Boolean constants can directly be used in conditional statements or require statements.

Therefore, it's not considered a better practice to explicitly use **TRUE or FALSE** in the **require** statements.

```
437         require(_blacklists[_address] == false,
438         _blacklists[_address] = true;
439         return true;
```

Recommendation:

The equality to boolean constants must be removed from the above-mentioned line.

A.2 External Visibility should be preferred
Explanation:

Those functions that are never called throughout the contract should be marked as ***external*** visibility instead of ***public*** visibility.

This will effectively result in Gas Optimization as well.

Therefore, the following function must be marked as **external** within the contract:

- ***name***
- ***symbol***
- ***decimals***
- ***totalSupply***
- ***balanceOf***
- ***paused***

- ***blackListed***
- ***pause***
- ***unpause***
- ***blacklist***
- ***whitelist***
- ***tansfer***
- ***mint***
- ***burn***

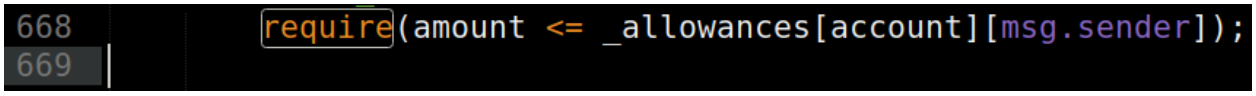
Recommendation:
If the **PUBLIC** visibility of the above-mentioned function is not intended, the function visibility should be changed to **EXTERNAL.**

A.3 Absence of Error messages in Require Statements

Line no - 668

Description:

The ***_burnFrom*** has a **require** statement in the QuiDax.sol contract that does not include an error message.



While this makes it troublesome to detect the reason behind a particular function revert, it also reduces the readability of the code.

Recommendation:

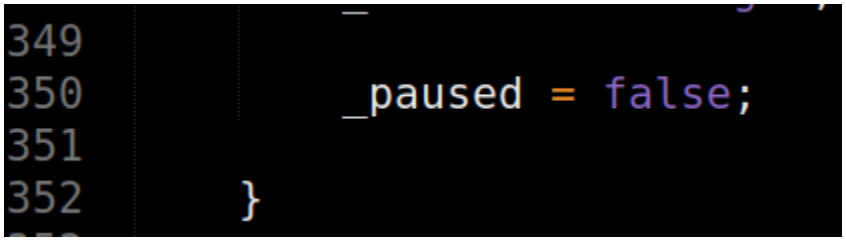
Error Messages must be included in every require statement in the contract.

A.7 Redundant Initialization of State Variable

Line no - 350

Description:

The State Variable **_paused** is being initialized to **FALSE** in the constructor.



However, boolean state variables are **FALSE** by default and do not require explicit initialization to false.

Recommendation:
Unnecessary Initialization of State variables should be avoided in the contract.

Informational

A.4 Coding Style Issues in the Contract

Explanation:
Code readability of a Smart Contract is largely influenced by the Coding Style issues and in some specific scenarios may lead to bugs in the future.

During the automated testing, it was found that the Quidax contract had quite a few code style issues.

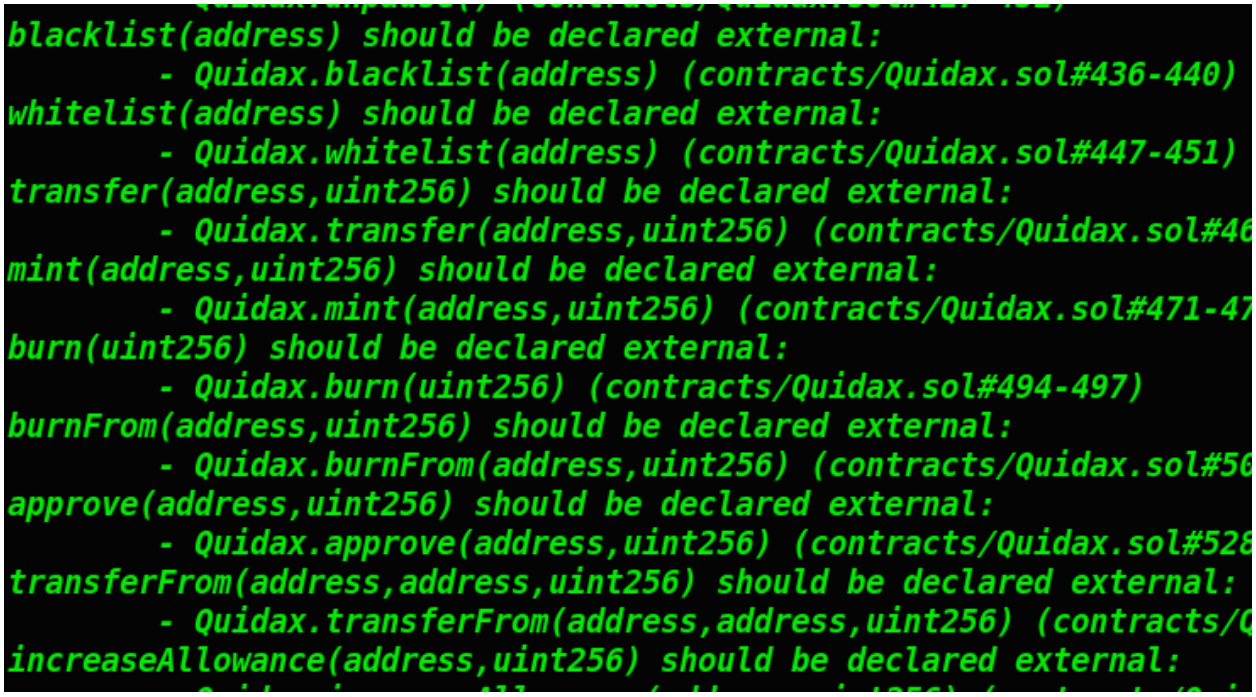
```
Parameter Quidax.blacklisted(address)._address (contracts/Quidax.sol#410) is not in mixedCase
Parameter Quidax.blacklist(address)._address (contracts/Quidax.sol#436) is not in mixedCase
Parameter Quidax.whitelist(address)._address (contracts/Quidax.sol#447) is not in mixedCase
Parameter Quidax.mintToMultipleAddresses(address[],uint256[])._addresses (contracts/Quidax.sol#479) is not in mixedCase
Parameter Quidax.mintToMultipleAddresses(address[],uint256[])._amount (contracts/Quidax.sol#479) is not in mixedCase
```

Recommendation:
Therefore, it is highly recommended to fix the issues like naming convention, indentation, and code layout issues in a smart contract.

Automated Test Results

```
Quidax.pause() (contracts/Quidax.sol#418-422) compares to a boolean constant:
        -require(bool,string)(_paused == false,Quidax: token transfer is unavailable) (contracts/Quidax.sol#41
Quidax.unpause() (contracts/Quidax.sol#427-431) compares to a boolean constant:
        -require(bool,string)(_paused == true,Quidax: token transfer is available) (contracts/Quidax.sol#428)
Quidax.blacklist(address) (contracts/Quidax.sol#436-440) compares to a boolean constant:
        -require(bool,string)(_blacklists[_address] == false,Quidax: account already blacklisted) (contracts/Q
Quidax.whitelist(address) (contracts/Quidax.sol#447-451) compares to a boolean constant:
        -require(bool,string)(_blacklists[_address] == true,Quidax: account already whitelisted) (contracts/Qu
Quidax._transfer(address,address,uint256) (contracts/Quidax.sol#602-614) compares to a boolean constant:
        -require(bool,string)(_blacklists[recipient] == false,Quidax: sender account already blacklisted) (con
Quidax._transfer(address,address,uint256) (contracts/Quidax.sol#602-614) compares to a boolean constant:
        -require(bool,string)(_blacklists[sender] == false,Quidax: sender account already blacklisted) (contra
Quidax._transfer(address,address,uint256) (contracts/Quidax.sol#602-614) compares to a boolean constant:
        -require(bool,string)(_paused == false,Quidax: token contract is not available) (contracts/Quidax.sol#
Quidax._mint(address,uint256) (contracts/Quidax.sol#625-635) compares to a boolean constant:
        -require(bool,string)(_paused == false,Quidax: token contract is not available) (contracts/Quidax.sol#
Quidax._mint(address,uint256) (contracts/Quidax.sol#625-635) compares to a boolean constant:
        -require(bool,string)(_blacklists[account] == false,Quidax: account to mint to already blacklisted) (c
Quidax._burn(address,uint256) (contracts/Quidax.sol#648-658) compares to a boolean constant:
        -require(bool,string)(_blacklists[account] == false,Quidax: account to burn from already blacklisted)
Quidax._burn(address,uint256) (contracts/Quidax.sol#648-658) compares to a boolean constant:
        -require(bool,string)(_paused == false,Quidax: token contract is not available) (contracts/Quidax.sol#
```

```
Quidax.constructor() (contracts/Quidax.sol#341-352) uses literals with too many digits:
        - _mint(_msgSender(),500000000000000000000000000) (contracts/Quidax.sol#346)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```


blacklist(address) should be declared external:
- Quidax.blacklist(address) (contracts/Quidax.sol#436-440)
whitelist(address) should be declared external:
- Quidax.whitelist(address) (contracts/Quidax.sol#447-451)
transfer(address,uint256) should be declared external:
- Quidax.transfer(address,uint256) (contracts/Quidax.sol#46
mint(address,uint256) should be declared external:
- Quidax.mint(address,uint256) (contracts/Quidax.sol#471-47
burn(uint256) should be declared external:
- Quidax.burn(uint256) (contracts/Quidax.sol#494-497)
burnFrom(address,uint256) should be declared external:
- Quidax.burnFrom(address,uint256) (contracts/Quidax.sol#50
approve(address,uint256) should be declared external:
- Quidax.approve(address,uint256) (contracts/Quidax.sol#528
transferFrom(address,address,uint256) should be declared external:
- Quidax.transferFrom(address,address,uint256) (contracts/Q
increaseAllowance(address,uint256) should be declared external: