QuillAudits

# Audit Report
# August, 2021

OpenExhibition

# Contents

## Scope of Audit

The scope of this audit was to analyze and document the OpenExhibitionToken Token smart contract codebase for quality, security, and correctness.

## Checked Vulnerabilities

We have scanned the smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that we considered:

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked math
- Unsafe type inference
- Implicit visibility level

# Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

**Structural Analysis**
In this step we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.
SmartCheck.

**Static Analysis**
Static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step a series of automated tools are used to test security of smart contracts.

**Code Review / Manual Analysis**
Manual Analysis or review of code was done to identify new vulnerability or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of automated analysis were manually verified.

**Gas Consumption**
In this step we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and possibilities of optimization of code to reduce gas consumption.

**Tools and Platforms used for Audit**
Remix IDE, Truffle, Truffle Team, Ganache, Solhint, Mythril, Slither, SmartCheck.

# Issue Categories

Every issue in this report has been assigned with a severity level. There are four levels of severity and each of them has been explained below.

## High severity issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality and we recommend these issues to be fixed before moving to a live environment.

## Medium level severity issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems and they should still be fixed.

## Low level severity issues

Low level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

## Informational

These are severity four issues which indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

## Number of issues per severity

| Type | High | Medium | Low | Informational |
|---|---|---|---|---|
| Open | 0 | 0 | 0 | 0 |
| Acknowledged | 0 | 3 | 4 | 6 |
| Closed | 0 | 2 | 2 | 7 |

# Introduction

During the period of **August 05, 2021 to August 12, 2021** - QuillAudits Team performed a security audit for OpenExhibitionToken smart contracts.

The code for the audit was taken from following the official link:

| Note | Date | Commit hash |
|---|---|---|
| Version 1 | August 05 | https://rinkeby.etherscan.io/address/0xBe19b47A55c996d40ec9a1fa423a518389344d93#code |
| Version 2 | August 12 | https://rinkeby.etherscan.io/address/0xeD15481475D397c9C90BD561BaCBA15E2a3E9009#code |

# Issues Found – Code Review / Manual Testing

## High severity issues

No issues were found.

## Medium severity issues

1. **Missing Check for Reentrancy Attack**

   **Description**
   Calling OpenExhibitionToken.transfer() and OpenExhibitionToken.transferFrom() might trigger function uniswapV2Router.addLiquidityETH , which is implemented by third party at IUniswapV2Router02. If there are vulnerable external calls in IUniswapV2Router02 , reentrancy attacks could be conducted because these two functions have state updates and event emits after external calls.

   The scope of the audit would treat the third-party implementation at IUniswapV2Router02 as a black box and assume its functional correctness. However, third parties may be compromised in the real world that leads to assets being lost or stolen.

   **Remediation**
   We recommend applying OpenZeppelin ReentrancyGuard library - nonReentrant modifier for the aforementioned functions to prevent reentrancy attacks.

   Status: Acknowledged by the Auditee

2. **Centralization Risks**

The role owner has the authority to update the critical settings

- setBurnPercent()
- setCashBackFeePercent()
- setCashBackPercent()
- setDevAccount()
- setDevFeePercent()
- setLiquidityFeePercent()

- setLiquifyEnabled()
- setMaxTxPercent()
- setMinTokenForLiquify()
- setNFTExchangeAddress()
- setTaxFeePercent()
- and other exclude and include from fees and rewards settings.

**Remediation**

We advise the Auditee to handle the governance account carefully to avoid any potential hack. We also advise the Auditee to consider the following solutions:

- with reasonable latency for community awareness on privileged operations;
- Multisig with community-voted 3rd-party independent co-signers;
- DAO or Governance module increasing transparency and community involvement;

**Status: Acknowledged by the Auditee**

### 3. Lack of event emissions

**Description**

The critical settings are completely devoid of event definitions or emissions. This makes it very difficult for users or other interested parties to track important changes that take place in the system.

Hence, the missing event makes it difficult to track off-chain liquidity fee changes. An event should be emitted for significant transactions calling the following functions:

- setBurnPercent()
- setCashBackFeePercent()
- setCashBackPercent()
- setDevAccount()
- setDevFeePercent()
- setLiquidityFeePercent()
- setLiquifyEnabled()

- setMaxTxPercent()
- setMinTokenForLiquify()
- setNFTExchangeAddress()
- setTaxFeePercent()
- excludeFromReward()
- includeInReward()
- excludeFromFee()
- excludeFromFeeAndRewards()
- includeInFee()

**Remediation**
Consider emitting events after sensitive changes take place to facilitate tracking and notify off-chain clients that may be following the contracts' activity.

As a result, we recommend logging the update of those variables that change when the aforementioned functions are called.

**Status:** Fixed

### 4. An amount of OEC is allowed for the uniswapV2Router

**Description**
An amount of OEC owned by the OpenExhibitionToken contract is allowed for the uniswapV2Router in the addLiquidity() function.

L759: _approve(address(this), address(uniswapV2Router), tokenAmount);

The scope of the audit would treat the third-party implementation at uniswapV2Router as a black box and assume its functional correctness. However, third parties may be compromised in the real world that leads to assets being lost or stolen.

**Remediation**
The Auditee should be aware of the aforementioned risk, which leads to lost or stolen assets and ensure that the uniswapV2Router is a trusted party during the deployment.

**Status: Acknowledged by the Auditee**

### 5. Loops in the contract are extremely costly

#### Description

The for loops in the entire codebase includes state variables .length of a non-memory array, in the condition of the for loops. As a result, these state variables consume a lot more extra gas for every iteration of the for loop.

#### Remediation

We recommend using a local variable instead of a state variable .length in a loop for the entire codebase.
E.g.: We should declare a local variable for _excluded.length before using it in the loop.

```
excludedLength = _excluded.length
for (uint256 i = 0; i < excludedLength; i++) {
}
```

**Status:** Fixed

## Low level severity issues

### 6. Redundant Code

| Line | Code |
|---|---|
| 786 | else if (!_isExcluded[sender] && !_isExcluded[recipient]) {<br>_transferStandard(sender, recipient, amount);<br>} |

#### Description

When the contract enters the branch else if (!_isExcluded[sender] && !_isExcluded[recipient]) or else, the contract will execute the same piece of code _transferStandard(sender, recipient, amount);

#### Remediation

We recommend removing the following code:

```
else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
        _transferStandard(sender, recipient, amount);
    }
```

**Status:** Acknowledged by the Auditee

## 7. Missing Range Check for Input Variables

### Description

The role can set the following state variables arbitrary large or small causing potential risks in fees and anti whale:

- setBurnPercent()
- setCashBackFeePercent()
- setCashBackPercent()
- setDevFeePercent()
- setLiquidityFeePercent()
- setMaxTxPercent()
- setMinTokenForLiquify()
- setTaxFeePercent()

### Remediation

We recommend setting ranges and check the following input variables:

- burnFee
- cashbackFee
- cashbackPercent
- devFee
- liquidityFee
- maxTxPercent
- amount
- taxFee

**Status:** Fixed

## 8. Lack of Zero address validation

### Description

To favor explicitness, consider adding a check for all functions that are taking address parameters in the entire codebase. Although most of the functions throughout the codebase properly validate function inputs, there are some instances of functions that do not. One example is: includeInReward - does not check for zero address

**Remediation**
Consider implementing require statements where appropriate to validate all user-controlled input, including governance functions, to avoid the potential for erroneous values to result in unexpected behaviors or wasted gas.

**Status: Acknowledged by the Auditee**

### 9. Lack of input validation

**Description**
The following functions should validate the address if it's already included or excluded:

- excludeFromFee()
- excludeFromFeeAndRewards()
- includeInFee()

Moreover, the function setLiquifyEnabled() should check whether _liquifyEnabled is enabled or disabled and revert if it's the same status.

**Remediation**
Consider implementing require statements where appropriate to validate all user-controlled input, including governance functions, to avoid the potential for erroneous values to result in unexpected behaviors or wasted gas.

**Status: Acknowledged by the Auditee**

### 10. Incorrect Error Message

**L438: require(_isExcluded[account], "Account is already included");**

**Description**
The error message in require(_isExcluded[account], "Account is already excluded") does not describe the error correctly.

**Remediation**
We recommend changing "Account is already excluded" to "Account is not excluded."

**Status:** Fixed

### 11. Avoid using tx.origin

#### Description

tx.origin is a global variable in Solidity that returns the account's address that sent the transaction. The contract does not use the variable for authorization. Still, if this is used in the next development for authorization purposes, it could make a contract vulnerable if an authorized account calls into a malicious contract.

A call could be made to the vulnerable contract that passes the authorization check since tx.origin returns the original sender of the transaction, which in this case is the authorized account.

#### Remediation

tx.origin should not be used. We recommend using msg.sender instead.

**Status: Acknowledged by the Auditee**

## Informational

### 12. Not using delete to zero values

In the removeAllFee function within the OpenExhibitionToken contract, the following variables are manually replaced with Zero:

- _taxFee = 0;
- _liquidityFee = 0;
- _devFee = 0;
- _cashbackFee = 0;
- _burnFee = 0;

To simplify the code and clarify intent, consider using delete instead.

**Status:** Fixed

### 13. Different pragma directives are used

Different versions of Solidity are used >=0.6.0<0.8.0 . >=0.6.2<0.8.0, ^0.6.12.

**Description**
solc frequently releases new compiler versions. Using an old version prevents access to new Solidity security checks. We also recommend avoiding complex pragma statements.

**Remediation**
Deploy with any of the following Solidity versions:
- 0.5.16 - 0.5.17
- 0.6.11 - 0.6.12
- 0.7.5 - 0.7.6

Use a simple pragma version that allows any of these versions. Consider using the latest version of Solidity for testing.

**Status:** Fixed

## 14. Solidity Private Modifier Do Not Hide Data

To favor explicitness, consider everything that is inside a contract is visible to all observers external to the blockchain. Therefore, the Auditee should be aware that making something private only prevents other contracts from reading or modifying the information, but it will still be visible to the whole world outside of the blockchain.

For example, the following private variables in OpenExhibitionToken contract are still visible and accessible:
- mapping (address => bool) private _isExcludedFromFee;
- mapping (address => bool) private _isExcluded;
- address[] private _excluded;
- uint256 private _tFeeTotal;

**Status: Acknowledged by the Auditee**

## 15. State variables that could be declared constant

Adding the constant attributes to state variables that never change. Therefore, the following constant state variables should be declared constant to save gas:
_tTotal
BURN_ADDRESS

**Status:** Fixed

### 16. Public function that could be declared external

Using the external attribute for functions never called from the contract. Therefore, the following public functions that are never called by the contract should be declared external to save gas:

- deliver()
- reflectionFromToken()
- excludeFromFeeAndRewards()
- includeInFee()
- setLiquifyEnabled()
- setMinTokenForLiquify()

**Status:** Fixed

### 17. State Variable Default Visibility

| Line | Code |
|---|---|
| 242 | address BURN_ADDRESS = 0x0000000000000000000000000000000000000001; |
| 281 | bool _liquify; |

**Description**

The Visibility of the above-mentioned variable is not defined. Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

The default is internal for state variables, but it should be made explicit.

**Remediation**

We recommend adding visibility for these variables.  Variables can be specified as being public, internal or private. Explicitly define visibility for all state variables.

**Status:** Fixed

### 18. Variables declared as uint instead of uint256

To favor explicitness, consider changing all instances of uint into uint256 in the entire codebase.

**Status:** Acknowledged by the Auditee

### 19. TODOs in code

At line 687 in the OpenExhibitionToken contract, there are “TODO” comments that should be removed and instead tracked in the project’s issues backlog.

**Status:** Fixed

### 20. Removing irrelevant comments

Irrelevant comments were found in the codebase, for example:
**L513:** // contract address will be always excluded.

We recommend removing those comments to increase the readability and quality of the code and ensure other irrelevant comments in other places are removed as well.

**Status: Acknowledged by the Auditee**

### 21. Misleading function and variable names

To favor explicitness and readability, some functions and variables from the whole codebase may benefit from a better naming.
Our suggestions are:
- getPathForETHtoOEC to getPathForETHToOEC
- calculateCashback to calculateNFTCashback
- setCashbackPercent to setNFTCashbackPercent

**Status:** Fixed

### 22. Inconsistent coding style

Deviations from the Solidity Style Guide were identified throughout the entire codebase. Taking into consideration how much value a consistent coding style adds to the project’s readability, enforcing a standard coding style with the help of linter tools such as Solhint is recommended.

**Status: Acknowledged by the Auditee**

### 23. block.timestamp may not be reliable

The Time contract uses the block.timestamp as part of the calculations and time checks.

Nevertheless, timestamps can be slightly altered by miners to favor them in contracts that have logics that depend strongly on them. Consider taking into account this issue and warning the users that such a scenario could happen.

**Status: Acknowledged by the Auditee**

### 24. Missing docstrings

It is extremely difficult to locate any contracts or functions, as they lack documentation. One consequence of this is that reviewers' understanding of the code's intention is impeded, which is significant because it is necessary to accurately determine both security and correctness.

They are additionally more readable and easier to maintain when wrapped in docstrings. The functions should be documented so that users can understand the purpose or intention of each function, as well as the situations in which it may fail, who is allowed to call it, what values it returns, and what events it emits.

**Status: Acknowledged by the Auditee**

# Functional test

| Function Names | Testing results |
|---|---|
| setBurnPercent() | Passed |
| setCashBackFeePercent() | Passed |
| setCashBackPercent() | Passed |
| setDevAccount() | Passed |
| setDevFeePercent() | Passed |
| setLiquidityFeePercent() | Passed |
| setLiquifyEnabled() | Passed |
| setMaxTxPercent() | Passed |
| setMinTokenForLiquify() | Passed |
| setNFTExchangeAddress() | Passed |
| setTaxFeePercent() | Passed |
| excludeFromReward() | Passed |
| includeInReward() | Passed |
| excludeFromFee() | Passed |
| excludeFromFeeAndRewards() | Passed |
| includeInFee() | Passed |
| increaseAllowance() | Passed |
| renounceOwnership() | Passed |
| transfer | Passed |
| transferFrom() | Passed |

| Function Names | Testing results |
| --- | --- |
| transferOwnerShip() | Passed |
| approve() | Passed |
| deliver() | Passed |
| decreaseAllowance() | Passed |
| withdrawXToken() | Passed |

# Automated Testing

## Slither

```
INFO:Detectors:
Reentrancy in OpenExhibitionToken._transfer(address,address,uint256) (OpenExhibitionToken.sol#709-755):
        External calls:
        - WETH.withdraw(wethBalance) (OpenExhibitionToken.sol#739)
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        External calls sending eth:
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _rOwned[sender] = _rOwned[sender].add(cbRAmount) (OpenExhibitionToken.sol#804)
                - _rOwned[address(this)] = _rOwned[address(this)].sub(cbRAmount) (OpenExhibitionToken.sol#805)
                - _rOwned[address(this)] = _rOwned[address(this)].add(rLiquidity) (OpenExhibitionToken.sol#597)
                - _rOwned[BURN_ADDRESS] = _rOwned[BURN_ADDRESS].add(rBurn) (OpenExhibitionToken.sol#630)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#819)
                - _rOwned[address(this)] = _rOwned[address(this)].add(rCashback) (OpenExhibitionToken.sol#620)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#832)
                - _rOwned[_devAccount] = _rOwned[_devAccount].add(rDev) (OpenExhibitionToken.sol#610)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#847)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#862)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#820)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#834)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#863)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#849)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _rTotal = _rTotal.sub(rFee) (OpenExhibitionToken.sol#524)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _rTotalForLiquidity = _rTotalForLiquidity.add(rLiquidity) (OpenExhibitionToken.sol#598)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _tOwned[sender] = _tOwned[sender].sub(tAmount) (OpenExhibitionToken.sol#846)
                - _tOwned[sender] = _tOwned[sender].sub(tAmount) (OpenExhibitionToken.sol#861)
                - _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount) (OpenExhibitionToken.sol#833)
                - _tOwned[address(this)] = _tOwned[address(this)].add(tCashbackFee) (OpenExhibitionToken.sol#622)
                - _tOwned[sender] = _tOwned[sender].add(cbTAmount) (OpenExhibitionToken.sol#810)
                - _tOwned[_devAccount] = _tOwned[_devAccount].add(tDevFee) (OpenExhibitionToken.sol#612)
                - _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount) (OpenExhibitionToken.sol#848)
                - _tOwned[BURN_ADDRESS] = _tOwned[BURN_ADDRESS].add(tBurn) (OpenExhibitionToken.sol#632)
                - _tOwned[address(this)] = _tOwned[address(this)].add(tLiquidity) (OpenExhibitionToken.sol#600)
                - _tOwned[address(this)] = _tOwned[address(this)].sub(cbTAmount) (OpenExhibitionToken.sol#812)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _tTotalForLiquidity = _tTotalForLiquidity.add(tLiquidity) (OpenExhibitionToken.sol#601)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities
INFO:Detectors:
```

```
INFO:Detectors:
OpenExhibitionToken._getValues(uint256).rTotalFees (OpenExhibitionToken.sol#530) is a local variable never initialized
OpenExhibitionToken._getValues(uint256).tTotalFees (OpenExhibitionToken.sol#529) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables
INFO:Detectors:
OpenExhibitionToken.addLiquidity(uint256,uint256) (OpenExhibitionToken.sol#757-775) ignores return value by uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owne
r(),block.timestamp) (OpenExhibitionToken.sol#762-769)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return
INFO:Detectors:
OpenExhibitionToken.allowance(address,address).owner (OpenExhibitionToken.sol#365) shadows:
        - Ownable.owner() (Ownable.sol#35-37) (function)
OpenExhibitionToken._approve(address,address,uint256).owner (OpenExhibitionToken.sol#701) shadows:
        - Ownable.owner() (Ownable.sol#35-37) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing
```

```
INFO:Detectors:
name() should be declared external:
        - OpenExhibitionToken.name() (OpenExhibitionToken.sol#339-341)
symbol() should be declared external:
        - OpenExhibitionToken.symbol() (OpenExhibitionToken.sol#343-345)
decimals() should be declared external:
        - OpenExhibitionToken.decimals() (OpenExhibitionToken.sol#347-349)
totalSupply() should be declared external:
        - OpenExhibitionToken.totalSupply() (OpenExhibitionToken.sol#351-353)
transfer(address,uint256) should be declared external:
        - OpenExhibitionToken.transfer(address,uint256) (OpenExhibitionToken.sol#360-363)
allowance(address,address) should be declared external:
        - OpenExhibitionToken.allowance(address,address) (OpenExhibitionToken.sol#365-367)
approve(address,uint256) should be declared external:
        - OpenExhibitionToken.approve(address,uint256) (OpenExhibitionToken.sol#369-372)
transferFrom(address,address,uint256) should be declared external:
        - OpenExhibitionToken.transferFrom(address,address,uint256) (OpenExhibitionToken.sol#374-378)
increaseAllowance(address,uint256) should be declared external:
        - OpenExhibitionToken.increaseAllowance(address,uint256) (OpenExhibitionToken.sol#380-383)
decreaseAllowance(address,uint256) should be declared external:
        - OpenExhibitionToken.decreaseAllowance(address,uint256) (OpenExhibitionToken.sol#385-388)
isExcludedFromReward(address) should be declared external:
        - OpenExhibitionToken.isExcludedFromReward(address) (OpenExhibitionToken.sol#390-392)
totalFees() should be declared external:
        - OpenExhibitionToken.totalFees() (OpenExhibitionToken.sol#394-396)
totalBurned() should be declared external:
        - OpenExhibitionToken.totalBurned() (OpenExhibitionToken.sol#398-400)
deliver(uint256) should be declared external:
        - OpenExhibitionToken.deliver(uint256) (OpenExhibitionToken.sol#402-409)
reflectionFromToken(uint256,bool) should be declared external:
        - OpenExhibitionToken.reflectionFromToken(uint256,bool) (OpenExhibitionToken.sol#411-420)
excludeFromFeeAndRewards(address) should be declared external:
        - OpenExhibitionToken.excludeFromFeeAndRewards(address) (OpenExhibitionToken.sol#466-469)
includeInFee(address) should be declared external:
        - OpenExhibitionToken.includeInFee(address) (OpenExhibitionToken.sol#471-473)
setLiquifyEnabled(bool) should be declared external:
        - OpenExhibitionToken.setLiquifyEnabled(bool) (OpenExhibitionToken.sol#503-506)
setMinTokenForLiquify(uint256) should be declared external:
        - OpenExhibitionToken.setMinTokenForLiquify(uint256) (OpenExhibitionToken.sol#508-510)
getCashbackBalance() should be declared external:
        - OpenExhibitionToken.getCashbackBalance() (OpenExhibitionToken.sol#512-514)
isExcludedFromFee(address) should be declared external:
        - OpenExhibitionToken.isExcludedFromFee(address) (OpenExhibitionToken.sol#683-685)
renounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (Ownable.sol#54-57)
transferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (Ownable.sol#63-67)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

```
Variable OpenExhibitionToken._transferStandard(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#818) is too similar to OpenExhibitionToken._transferBothExcluded(address,addres
s,uint256).tTransferAmount (OpenExhibitionToken.sol#845)
Variable OpenExhibitionToken._transferToExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#831) is too similar to OpenExhibitionToken._transferToExcluded(address,addres
s,uint256).tTransferAmount (OpenExhibitionToken.sol#831)
Variable OpenExhibitionToken._transferToExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#831) is too similar to OpenExhibitionToken._transferFromExcluded(address,addr
ess,uint256).tTransferAmount (OpenExhibitionToken.sol#860)
Variable OpenExhibitionToken._transferToExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#831) is too similar to OpenExhibitionToken._transferBothExcluded(address,addr
ess,uint256).tTransferAmount (OpenExhibitionToken.sol#845)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._transferToExcluded(address,addr
ess,uint256).tTransferAmount (OpenExhibitionToken.sol#831)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar
INFO:Detectors:
OpenExhibitionToken.slitherConstructorVariables() (OpenExhibitionToken.sol#229-873) uses literals with too many digits:
        - BURN_ADDRESS = 0x0000000000000000000000000000000000000001 (OpenExhibitionToken.sol#242)
OpenExhibitionToken.slitherConstructorVariables() (OpenExhibitionToken.sol#229-873) uses literals with too many digits:
        - _tTotal = 1000000000 * 10 ** 6 * 10 ** 9 (OpenExhibitionToken.sol#245)
OpenExhibitionToken.slitherConstructorVariables() (OpenExhibitionToken.sol#229-873) uses literals with too many digits:
        - _maxTxAmount = 5000000 * 10 ** 6 * 10 ** 9 (OpenExhibitionToken.sol#284)
OpenExhibitionToken.slitherConstructorVariables() (OpenExhibitionToken.sol#229-873) uses literals with too many digits:
        - _minTokensToAddLiquidity = 500000 * 10 ** 6 * 10 ** 9 (OpenExhibitionToken.sol#285)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
INFO:Detectors:
OpenExhibitionToken.BURN_ADDRESS (OpenExhibitionToken.sol#242) should be constant
OpenExhibitionToken._decimals (OpenExhibitionToken.sol#251) should be constant
OpenExhibitionToken._name (OpenExhibitionToken.sol#249) should be constant
OpenExhibitionToken._symbol (OpenExhibitionToken.sol#250) should be constant
OpenExhibitionToken._tTotal (OpenExhibitionToken.sol#245) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant
INFO:Detectors:
name() should be declared external:
        - OpenExhibitionToken.name() (OpenExhibitionToken.sol#339-341)
symbol() should be declared external:
        - OpenExhibitionToken.symbol() (OpenExhibitionToken.sol#343-345)
decimals() should be declared external:
        - OpenExhibitionToken.decimals() (OpenExhibitionToken.sol#347-349)
totalSupply() should be declared external:
        - OpenExhibitionToken.totalSupply() (OpenExhibitionToken.sol#351-353)
transfer(address,uint256) should be declared external:
        - OpenExhibitionToken.transfer(address,uint256) (OpenExhibitionToken.sol#360-363)
allowance(address,address) should be declared external:
        - OpenExhibitionToken.allowance(address,address) (OpenExhibitionToken.sol#365-367)
approve(address,uint256) should be declared external:
        - OpenExhibitionToken.approve(address,uint256) (OpenExhibitionToken.sol#369-372)
transferFrom(address,address,uint256) should be declared external:
        - OpenExhibitionToken.transferFrom(address,address,uint256) (OpenExhibitionToken.sol#374-378)
increaseAllowance(address,uint256) should be declared external:
        - OpenExhibitionToken.increaseAllowance(address,uint256) (OpenExhibitionToken.sol#380-383)
decreaseAllowance(address,uint256) should be declared external:
        - OpenExhibitionToken.decreaseAllowance(address,uint256) (OpenExhibitionToken.sol#385-388)
isExcludedFromReward(address) should be declared external:
```

```
unt (OpenExhibitionToken.sol#860)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._transferStandard(address,addres
s,uint256).tTransferAmount (OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._getValues(uint256).rTransferAmount (OpenExhibitionToken.sol#571) is too similar to OpenExhibitionToken._transferStandard(address,address,uint256).tTransferAmount
(OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._transferBothExcluded(address,ad
dress,uint256).tTransferAmount (OpenExhibitionToken.sol#845)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._getValues(uint256).tTransferAmo
unt (OpenExhibitionToken.sol#570)
Variable OpenExhibitionToken.reflectionFromToken(uint256,bool).rTransferAmount (OpenExhibitionToken.sol#417) is too similar to OpenExhibitionToken._transferFromExcluded(address,address,uint256
).tTransferAmount (OpenExhibitionToken.sol#860)
Variable OpenExhibitionToken._transferBothExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#845) is too similar to OpenExhibitionToken._getValues(uint256).tTransferAmo
unt (OpenExhibitionToken.sol#570)
Variable OpenExhibitionToken._transferBothExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#845) is too similar to OpenExhibitionToken._transferStandard(address,addres
s,uint256).tTransferAmount (OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._getValues(uint256).rTransferAmount (OpenExhibitionToken.sol#571) is too similar to OpenExhibitionToken._transferBothExcluded(address,address,uint256).tTransferAmo
unt (OpenExhibitionToken.sol#845)
Variable OpenExhibitionToken.reflectionFromToken(uint256,bool).rTransferAmount (OpenExhibitionToken.sol#417) is too similar to OpenExhibitionToken._transferStandard(address,address,uint256).tT
ransferAmount (OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._transferStandard(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#818) is too similar to OpenExhibitionToken._getValues(uint256).tTransferAmount
(OpenExhibitionToken.sol#570)
Variable OpenExhibitionToken.reflectionFromToken(uint256,bool).rTransferAmount (OpenExhibitionToken.sol#417) is too similar to OpenExhibitionToken._transferBothExcluded(address,address,uint256
).tTransferAmount (OpenExhibitionToken.sol#845)
Variable OpenExhibitionToken.reflectionFromToken(uint256,bool).rTransferAmount (OpenExhibitionToken.sol#417) is too similar to OpenExhibitionToken._transferToExcluded(address,address,uint256).
tTransferAmount (OpenExhibitionToken.sol#831)
Variable OpenExhibitionToken._getValues(uint256).rTransferAmount (OpenExhibitionToken.sol#571) is too similar to OpenExhibitionToken._getValues(uint256).tTransferAmount (OpenExhibitionToken.so
l#570)
Variable OpenExhibitionToken._transferBothExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#845) is too similar to OpenExhibitionToken._transferBothExcluded(address,ad
dress,uint256).tTransferAmount (OpenExhibitionToken.sol#845)
Variable OpenExhibitionToken.reflectionFromToken(uint256,bool).rTransferAmount (OpenExhibitionToken.sol#417) is too similar to OpenExhibitionToken._getValues(uint256).tTransferAmount (OpenExhi
bitionToken.sol#570)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._transferFromExcluded(address,ad
dress,uint256).tTransferAmount (OpenExhibitionToken.sol#860)
Variable OpenExhibitionToken._transferStandard(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#818) is too similar to OpenExhibitionToken._transferStandard(address,address,ui
nt256).tTransferAmount (OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._transferBothExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#845) is too similar to OpenExhibitionToken._transferFromExcluded(address,ad
dress,uint256).tTransferAmount (OpenExhibitionToken.sol#860)
Variable OpenExhibitionToken._transferToExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#831) is too similar to OpenExhibitionToken._transferStandard(address,address,
uint256).tTransferAmount (OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._transferToExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#831) is too similar to OpenExhibitionToken._getValues(uint256).tTransferAmoun
t (OpenExhibitionToken.sol#570)
Variable OpenExhibitionToken._getValues(uint256).rTransferAmount (OpenExhibitionToken.sol#571) is too similar to OpenExhibitionToken._transferToExcluded(address,address,uint256).tTransferAmoun
t (OpenExhibitionToken.sol#831)
Variable OpenExhibitionToken._transferBothExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#845) is too similar to OpenExhibitionToken._transferToExcluded(address,addr
ess,uint256).tTransferAmount (OpenExhibitionToken.sol#831)
Variable OpenExhibitionToken._transferStandard(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#818) is too similar to OpenExhibitionToken._transferFromExcluded(address,addres
s,uint256).tTransferAmount (OpenExhibitionToken.sol#860)
Variable OpenExhibitionToken._transferStandard(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#818) is too similar to OpenExhibitionToken._transferToExcluded(address,address,
uint256).tTransferAmount (OpenExhibitionToken.sol#831)
Variable OpenExhibitionToken._transferStandard(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#818) is too similar to OpenExhibitionToken._transferBothExcluded(address,addres
```

```
                - _approve(address(this),address(uniswapV2Router),tokenAmount) (OpenExhibitionToken.sol#759)
Reentrancy in OpenExhibitionToken.addLiquidity(uint256,uint256) (OpenExhibitionToken.sol#757-775):
        External calls:
        - refundGas() (OpenExhibitionToken.sol#757)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        External calls sending eth:
        - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        - refundGas() (OpenExhibitionToken.sol#757)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _rTotalForLiquidity = _rTotalForLiquidity.sub(rTokenAmount) (OpenExhibitionToken.sol#772)
        - _tTotalForLiquidity = _tTotalForLiquidity.sub(tokenAmount) (OpenExhibitionToken.sol#773)
        Event emitted after the call(s):
        - Liquify(tokenAmount,ethAmount) (OpenExhibitionToken.sol#774)
Reentrancy in OpenExhibitionToken.refundGas() (OpenExhibitionToken.sol#290-299):
        External calls:
        - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _liquify = false (OpenExhibitionToken.sol#298)
Reentrancy in OpenExhibitionToken.transferFrom(address,address,uint256) (OpenExhibitionToken.sol#374-378):
        External calls:
        - _transfer(sender,recipient,amount) (OpenExhibitionToken.sol#375)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (OpenExhibitionToken.sol#375)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (OpenExhibitionToken.sol#376)
                - _allowances[owner][spender] = amount (OpenExhibitionToken.sol#705)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (OpenExhibitionToken.sol#706)
                - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (OpenExhibitionToken.sol#376)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-4
INFO:Detectors:
Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountADesired (OpenExhibitionToken.sol#92) is too similar to IUniswapV2Router01.addLi
quidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (OpenExhibitionToken.sol#93)
Variable OpenExhibitionToken._cashbackFee (OpenExhibitionToken.sol#268) is too similar to OpenExhibitionToken._takeCashbackFee(uint256).tCashbackFee (OpenExhibitionToken.sol#615)
Variable OpenExhibitionToken._rTotalForLiquidity (OpenExhibitionToken.sol#312) is too similar to OpenExhibitionToken._tTotalForLiquidity (OpenExhibitionToken.sol#311)
Variable OpenExhibitionToken._getValues(uint256).rTransferAmount (OpenExhibitionToken.sol#571) is too similar to OpenExhibitionToken._transferFromExcluded(address,address,uint256).tTransferAmo
unt (OpenExhibitionToken.sol#860)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._transferStandard(address,addres
s,uint256).tTransferAmount (OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._getValues(uint256).rTransferAmount (OpenExhibitionToken.sol#571) is too similar to OpenExhibitionToken._transferStandard(address,address,uint256).tTransferAmount
(OpenExhibitionToken.sol#818)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._transferBothExcluded(address,ad
dress,uint256).tTransferAmount (OpenExhibitionToken.sol#845)
Variable OpenExhibitionToken._transferFromExcluded(address,address,uint256).rTransferAmount (OpenExhibitionToken.sol#860) is too similar to OpenExhibitionToken._getValues(uint256).tTransferAmo
unt (OpenExhibitionToken.sol#570)
Variable OpenExhibitionToken.reflectionFromToken(uint256,bool).rTransferAmount (OpenExhibitionToken.sol#417) is too similar to OpenExhibitionToken._transferFromExcluded(address,address,uint256
```

```
                - _rOwned[address(this)] = _rOwned[address(this)].sub(cbRAmount) (OpenExhibitionToken.sol#805)
                - _rOwned[address(this)] = _rOwned[address(this)].add(rLiquidity) (OpenExhibitionToken.sol#597)
                - _rOwned[BURN_ADDRESS] = _rOwned[BURN_ADDRESS].add(rBurn) (OpenExhibitionToken.sol#630)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#819)
                - _rOwned[address(this)] = _rOwned[address(this)].add(rCashback) (OpenExhibitionToken.sol#620)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#832)
                - _rOwned[_devAccount] = _rOwned[_devAccount].add(rDev) (OpenExhibitionToken.sol#610)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#847)
                - _rOwned[sender] = _rOwned[sender].sub(rAmount) (OpenExhibitionToken.sol#862)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#820)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#834)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#863)
                - _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount) (OpenExhibitionToken.sol#849)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _rTotal = _rTotal.sub(rFee) (OpenExhibitionToken.sol#524)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _rTotalForLiquidity = _rTotalForLiquidity.add(rLiquidity) (OpenExhibitionToken.sol#598)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _tFeeTotal = _tFeeTotal.add(tFee) (OpenExhibitionToken.sol#525)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _tOwned[sender] = _tOwned[sender].sub(tAmount) (OpenExhibitionToken.sol#846)
                - _tOwned[sender] = _tOwned[sender].sub(tAmount) (OpenExhibitionToken.sol#861)
                - _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount) (OpenExhibitionToken.sol#833)
                - _tOwned[address(this)] = _tOwned[address(this)].add(tCashbackFee) (OpenExhibitionToken.sol#622)
                - _tOwned[sender] = _tOwned[sender].add(cbTAmount) (OpenExhibitionToken.sol#810)
                - _tOwned[_devAccount] = _tOwned[_devAccount].add(tDevFee) (OpenExhibitionToken.sol#612)
                - _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount) (OpenExhibitionToken.sol#848)
                - _tOwned[BURN_ADDRESS] = _tOwned[BURN_ADDRESS].add(tBurn) (OpenExhibitionToken.sol#632)
                - _tOwned[address(this)] = _tOwned[address(this)].add(tLiquidity) (OpenExhibitionToken.sol#600)
                - _tOwned[address(this)] = _tOwned[address(this)].sub(cbTAmount) (OpenExhibitionToken.sol#812)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _tTotalForLiquidity = _tTotalForLiquidity.add(tLiquidity) (OpenExhibitionToken.sol#601)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _taxFee = _previousTaxFee (OpenExhibitionToken.sol#676)
                - _taxFee = 0 (OpenExhibitionToken.sol#668)
        Event emitted after the call(s):
        - Transfer(sender,recipient,tTransferAmount) (OpenExhibitionToken.sol#827)
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
        - Transfer(sender,recipient,tTransferAmount) (OpenExhibitionToken.sol#870)
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
        - Transfer(sender,recipient,tTransferAmount) (OpenExhibitionToken.sol#841)
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
        - Transfer(sender,recipient,tTransferAmount) (OpenExhibitionToken.sol#856)
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
Reentrancy in OpenExhibitionToken.addLiquidity(uint256,uint256) (OpenExhibitionToken.sol#757-775):
        External calls:
        - refundGas() (OpenExhibitionToken.sol#757)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _approve(address(this),address(uniswapV2Router),tokenAmount) (OpenExhibitionToken.sol#759)
```

```
INFO:Detectors:
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (OpenExhibitionToken.sol#47) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (OpenExhibitionToken.sol#48) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (OpenExhibitionToken.sol#65) is not in mixedCase
Function IUniswapV2Router01.WETH() (OpenExhibitionToken.sol#87) is not in mixedCase
Parameter OpenExhibitionToken.calculateTaxFee(uint256)._amount (OpenExhibitionToken.sol#635) is not in mixedCase
Parameter OpenExhibitionToken.calculateLiquidityFee(uint256)._amount (OpenExhibitionToken.sol#639) is not in mixedCase
Parameter OpenExhibitionToken.calculateDevFee(uint256)._amount (OpenExhibitionToken.sol#643) is not in mixedCase
Parameter OpenExhibitionToken.calculateCashbackFee(uint256)._amount (OpenExhibitionToken.sol#647) is not in mixedCase
Parameter OpenExhibitionToken.calculateBurnFee(uint256)._amount (OpenExhibitionToken.sol#651) is not in mixedCase
Variable OpenExhibitionToken.BURN_ADDRESS (OpenExhibitionToken.sol#242) is not in mixedCase
Variable OpenExhibitionToken._taxFee (OpenExhibitionToken.sol#262) is not in mixedCase
Variable OpenExhibitionToken._liquidityFee (OpenExhibitionToken.sol#265) is not in mixedCase
Variable OpenExhibitionToken._cashbackFee (OpenExhibitionToken.sol#268) is not in mixedCase
Variable OpenExhibitionToken._burnFee (OpenExhibitionToken.sol#271) is not in mixedCase
Variable OpenExhibitionToken._devFee (OpenExhibitionToken.sol#274) is not in mixedCase
Variable OpenExhibitionToken.WETH (OpenExhibitionToken.sol#279) is not in mixedCase
Variable OpenExhibitionToken._liquify (OpenExhibitionToken.sol#281) is not in mixedCase
Variable OpenExhibitionToken._liquifyEnabled (OpenExhibitionToken.sol#282) is not in mixedCase
Variable OpenExhibitionToken._maxTxAmount (OpenExhibitionToken.sol#284) is not in mixedCase
Variable OpenExhibitionToken._cashbackPercentage (OpenExhibitionToken.sol#301) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
INFO:Detectors:
Redundant expression "this (Context.sol#21)" inContext (Context.sol#15-25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
INFO:Detectors:
Reentrancy in OpenExhibitionToken._transfer(address,address,uint256) (OpenExhibitionToken.sol#709-755):
        External calls:
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        External calls sending eth:
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _burnFee = _previousBurnFee (OpenExhibitionToken.sol#680)
                - _burnFee = 0 (OpenExhibitionToken.sol#672)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _cashbackFee = _previousCashbackFee (OpenExhibitionToken.sol#679)
                - _cashbackFee = 0 (OpenExhibitionToken.sol#671)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _devFee = _previousDevFee (OpenExhibitionToken.sol#678)
                - _devFee = 0 (OpenExhibitionToken.sol#670)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _liquidityFee = _previousLiquidityFee (OpenExhibitionToken.sol#677)
                - _liquidityFee = 0 (OpenExhibitionToken.sol#669)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _previousBurnFee = _burnFee (OpenExhibitionToken.sol#666)
Address.functionDelegateCall(address,bytes) (Address.sol#153-155) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (Address.sol#163-169) is never used and should be removed
Address.functionStaticCall(address,bytes) (Address.sol#129-131) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (Address.sol#139-145) is never used and should be removed
Address.isContract(address) (Address.sol#26-35) is never used and should be removed
Address.sendValue(address,uint256) (Address.sol#53-59) is never used and should be removed
Context._msgData() (Context.sol#20-23) is never used and should be removed
SafeMath.div(uint256,uint256,string) (SafeMath.sol#190-193) is never used and should be removed
SafeMath.mod(uint256,uint256) (SafeMath.sol#152-155) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (SafeMath.sol#210-213) is never used and should be removed
SafeMath.tryAdd(uint256,uint256) (SafeMath.sol#24-28) is never used and should be removed
SafeMath.tryDiv(uint256,uint256) (SafeMath.sol#60-63) is never used and should be removed
SafeMath.tryMod(uint256,uint256) (SafeMath.sol#70-73) is never used and should be removed
SafeMath.tryMul(uint256,uint256) (SafeMath.sol#45-53) is never used and should be removed
SafeMath.trySub(uint256,uint256) (SafeMath.sol#35-38) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
INFO:Detectors:
OpenExhibitionToken._rTotal (OpenExhibitionToken.sol#246) is set pre-construction with a non-constant function or state variable:
        - (MAX - (MAX % _tTotal))
OpenExhibitionToken._previousTaxFee (OpenExhibitionToken.sol#263) is set pre-construction with a non-constant function or state variable:
        - _taxFee
OpenExhibitionToken._previousLiquidityFee (OpenExhibitionToken.sol#266) is set pre-construction with a non-constant function or state variable:
        - _liquidityFee
OpenExhibitionToken._previousCashbackFee (OpenExhibitionToken.sol#269) is set pre-construction with a non-constant function or state variable:
        - _cashbackFee
OpenExhibitionToken._previousBurnFee (OpenExhibitionToken.sol#272) is set pre-construction with a non-constant function or state variable:
        - _burnFee
OpenExhibitionToken._previousDevFee (OpenExhibitionToken.sol#275) is set pre-construction with a non-constant function or state variable:
        - _devFee
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state-variables
INFO:Detectors:
Pragma version>=0.6.2<0.8.0 (Address.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (Context.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (IERC20.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (Ownable.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (ReentrancyGuard.sol#3) is too complex
Pragma version>=0.6.0<0.8.0 (SafeMath.sol#3) is too complex
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
INFO:Detectors:
Low level call in Address.sendValue(address,uint256) (Address.sol#53-59):
        - (success) = recipient.call{value: amount}() (Address.sol#57)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (Address.sol#114-121):
        - (success,returndata) = target.call{value: value}(data) (Address.sol#119)
Low level call in Address.functionStaticCall(address,bytes,string) (Address.sol#139-145):
        - (success,returndata) = target.staticcall(data) (Address.sol#143)
Low level call in Address.functionDelegateCall(address,bytes,string) (Address.sol#163-169):
        - (success,returndata) = target.delegatecall(data) (Address.sol#167)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
INFO:Detectors:
```

```
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
Reentrancy in OpenExhibitionToken.addLiquidity(uint256,uint256) (OpenExhibitionToken.sol#757-775):
        External calls:
        - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        External calls sending eth:
        - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        - refundGas() (OpenExhibitionToken.sol#757)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        Event emitted after the call(s):
        - Liquify(tokenAmount,ethAmount) (OpenExhibitionToken.sol#774)
Reentrancy in OpenExhibitionToken.constructor(address) (OpenExhibitionToken.sol#314-337):
        External calls:
        - uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (OpenExhibitionToken.sol#322-323)
        Event emitted after the call(s):
        - Transfer(address(0),_msgSender(),_tTotal) (OpenExhibitionToken.sol#336)
Reentrancy in OpenExhibitionToken.transferFrom(address,address,uint256) (OpenExhibitionToken.sol#374-378):
        External calls:
        - _transfer(sender,recipient,amount) (OpenExhibitionToken.sol#375)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - WETH.withdraw(wethBalance) (OpenExhibitionToken.sol#739)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (OpenExhibitionToken.sol#375)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (OpenExhibitionToken.sol#706)
                - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (OpenExhibitionToken.sol#376)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
INFO:Detectors:
Address.isContract(address) (Address.sol#26-35) uses assembly
        - INLINE ASM (Address.sol#33)
Address._verifyCallResult(bool,bytes,string) (Address.sol#171-188) uses assembly
        - INLINE ASM (Address.sol#180-183)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
INFO:Detectors:
Different versions of Solidity is used:
        - Version used: ['>=0.6.0<0.8.0', '>=0.6.2<0.8.0', '^0.6.12']
        - >=0.6.2<0.8.0 (Address.sol#3)
        - >=0.6.0<0.8.0 (Context.sol#3)
        - >=0.6.0<0.8.0 (IERC20.sol#3)
        - ^0.6.12 (OpenExhibitionToken.sol#1)
        - >=0.6.0<0.8.0 (Ownable.sol#3)
        - >=0.6.0<0.8.0 (ReentrancyGuard.sol#3)
        - >=0.6.0<0.8.0 (SafeMath.sol#3)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used
INFO:Detectors:
Address._verifyCallResult(bool,bytes,string) (Address.sol#171-188) is never used and should be removed
Address.functionCall(address,bytes) (Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (Address.sol#89-91) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (Address.sol#104-106) is never used and should be removed
```

```
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _rTotalForLiquidity = _rTotalForLiquidity.sub(rTokenAmount) (OpenExhibitionToken.sol#772)
        - _tTotalForLiquidity = _tTotalForLiquidity.sub(tokenAmount) (OpenExhibitionToken.sol#773)
Reentrancy in OpenExhibitionToken.constructor(address) (OpenExhibitionToken.sol#314-337):
        External calls:
        - uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this),_uniswapV2Router.WETH()) (OpenExhibitionToken.sol#322-323)
        State variables written after the call(s):
        - WETH = IWETH(_uniswapV2Router.WETH()) (OpenExhibitionToken.sol#327)
        - _isExcluded[address(this)] = true (OpenExhibitionToken.sol#333)
        - _isExcluded[BURN_ADDRESS] = true (OpenExhibitionToken.sol#334)
        - _isExcludedFromFee[_msgSender()] = true (OpenExhibitionToken.sol#328)
        - _isExcludedFromFee[address(this)] = true (OpenExhibitionToken.sol#329)
        - _isExcludedFromFee[_devAccount] = true (OpenExhibitionToken.sol#330)
        - _isExcludedFromFee[BURN_ADDRESS] = true (OpenExhibitionToken.sol#331)
        - uniswapV2Router = _uniswapV2Router (OpenExhibitionToken.sol#326)
Reentrancy in OpenExhibitionToken.transferFrom(address,address,uint256) (OpenExhibitionToken.sol#374-378):
        External calls:
        - _transfer(sender,recipient,amount) (OpenExhibitionToken.sol#375)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - WETH.withdraw(wethBalance) (OpenExhibitionToken.sol#739)
        External calls sending eth:
        - _transfer(sender,recipient,amount) (OpenExhibitionToken.sol#375)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - _approve(sender,_msgSender(),_allowances[sender][_msgSender()].sub(amount,ERC20: transfer amount exceeds allowance)) (OpenExhibitionToken.sol#376)
                - _allowances[owner][spender] = amount (OpenExhibitionToken.sol#705)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2
INFO:Detectors:
Reentrancy in OpenExhibitionToken._transfer(address,address,uint256) (OpenExhibitionToken.sol#709-755):
        External calls:
        - WETH.withdraw(wethBalance) (OpenExhibitionToken.sol#739)
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        External calls sending eth:
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        Event emitted after the call(s):
        - Approval(owner,spender,amount) (OpenExhibitionToken.sol#706)
                - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
        - Liquify(tokenAmount,ethAmount) (OpenExhibitionToken.sol#774)
                - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
        - Transfer(sender,recipient,tTransferAmount) (OpenExhibitionToken.sol#827)
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
        - Transfer(sender,recipient,tTransferAmount) (OpenExhibitionToken.sol#870)
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
        - Transfer(sender,recipient,tTransferAmount) (OpenExhibitionToken.sol#841)
                - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
```

```
INFO:Detectors:
Reentrancy in OpenExhibitionToken._transfer(address,address,uint256) (OpenExhibitionToken.sol#709-755):
        External calls:
        - WETH.withdraw(wethBalance) (OpenExhibitionToken.sol#739)
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        External calls sending eth:
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
        State variables written after the call(s):
        - addLiquidity(liquidityBalance,eth) (OpenExhibitionToken.sol#742)
                - _allowances[owner][spender] = amount (OpenExhibitionToken.sol#705)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _burnFee = _previousBurnFee (OpenExhibitionToken.sol#680)
                - _burnFee = 0 (OpenExhibitionToken.sol#672)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _cashbackFee = _previousCashbackFee (OpenExhibitionToken.sol#679)
                - _cashbackFee = 0 (OpenExhibitionToken.sol#671)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _devFee = _previousDevFee (OpenExhibitionToken.sol#678)
                - _devFee = 0 (OpenExhibitionToken.sol#670)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _liquidityFee = _previousLiquidityFee (OpenExhibitionToken.sol#677)
                - _liquidityFee = 0 (OpenExhibitionToken.sol#669)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _previousBurnFee = _burnFee (OpenExhibitionToken.sol#666)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _previousCashbackFee = _cashbackFee (OpenExhibitionToken.sol#664)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _previousDevFee = _devFee (OpenExhibitionToken.sol#665)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _previousLiquidityFee = _liquidityFee (OpenExhibitionToken.sol#663)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _previousTaxFee = _taxFee (OpenExhibitionToken.sol#662)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _tFeeTotal = _tFeeTotal.add(tFee) (OpenExhibitionToken.sol#525)
        - _tokenTransfer(from,to,amount,takeFee) (OpenExhibitionToken.sol#754)
                - _taxFee = _previousTaxFee (OpenExhibitionToken.sol#676)
                - _taxFee = 0 (OpenExhibitionToken.sol#668)
Reentrancy in OpenExhibitionToken.addLiquidity(uint256,uint256) (OpenExhibitionToken.sol#757-775):
        External calls:
        - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        External calls sending eth:
        - uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this),tokenAmount,0,0,owner(),block.timestamp) (OpenExhibitionToken.sol#762-769)
        - refundGas() (OpenExhibitionToken.sol#757)
                - tx.origin.transfer(refundETH) (OpenExhibitionToken.sol#297)
```

## Mythril

```
enderphan@enderphan Open % myth a OpenExhibitionToken.sol
The analysis was completed successfully. No issues were detected.
```

## SOLHINT LINTER

```
OpenExhibitionToken.sol
   1:1    error    Compiler version ^0.6.12 does not satisfy the ^0.5.8 semver requirement  compiler-version
  47:5    warning  Function name must be in mixedCase                                       func-name-mixedcase
  48:5    warning  Function name must be in mixedCase                                       func-name-mixedcase
  65:5    warning  Function name must be in mixedCase                                       func-name-mixedcase
  87:5    warning  Function name must be in mixedCase                                       func-name-mixedcase
 229:1    warning  Contract has 32 states declarations but allowed no more than 15          max-states-count
 242:5    warning  Explicitly mark visibility of state                                      state-visibility
 242:5    warning  Variable name must be in mixedCase                                       var-name-mixedcase
 279:5    warning  Variable name must be in mixedCase                                       var-name-mixedcase
 281:5    warning  Explicitly mark visibility of state                                      state-visibility
 297:13   warning  Avoid to use tx.origin                                                   avoid-tx-origin
 404:9    warning  Error message for require is too long                                    reason-string
 423:9    warning  Error message for require is too long                                    reason-string
 457:9    warning  Error message for require is too long                                    reason-string
 521:32   warning  Code contains empty blocks                                              no-empty-blocks
 702:9    warning  Error message for require is too long                                    reason-string
 703:9    warning  Error message for require is too long                                    reason-string
 714:9    warning  Error message for require is too long                                    reason-string
 715:9    warning  Error message for require is too long                                    reason-string
 716:9    warning  Error message for require is too long                                    reason-string
 718:13   warning  Error message for require is too long                                    reason-string
 768:13   warning  Avoid to make time-based decisions in your business logic               not-rely-on-time
```

**Results**

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

# Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the OpenExhibitionToken platform. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the OpenExhibitionToken Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

# Closing Summary

In this report, we have considered the security of the OpenExhibitionToken platform. We performed our audit according to the procedure described above.

The audit showed several high, medium, low, and informational severity issues. In the end, the majority of the issues were fixed or acknowledged by the Auditee. Other issues still remained unfixed as the internal team needs further discussion.


OpenExhibition

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

audits@quillhash.com