

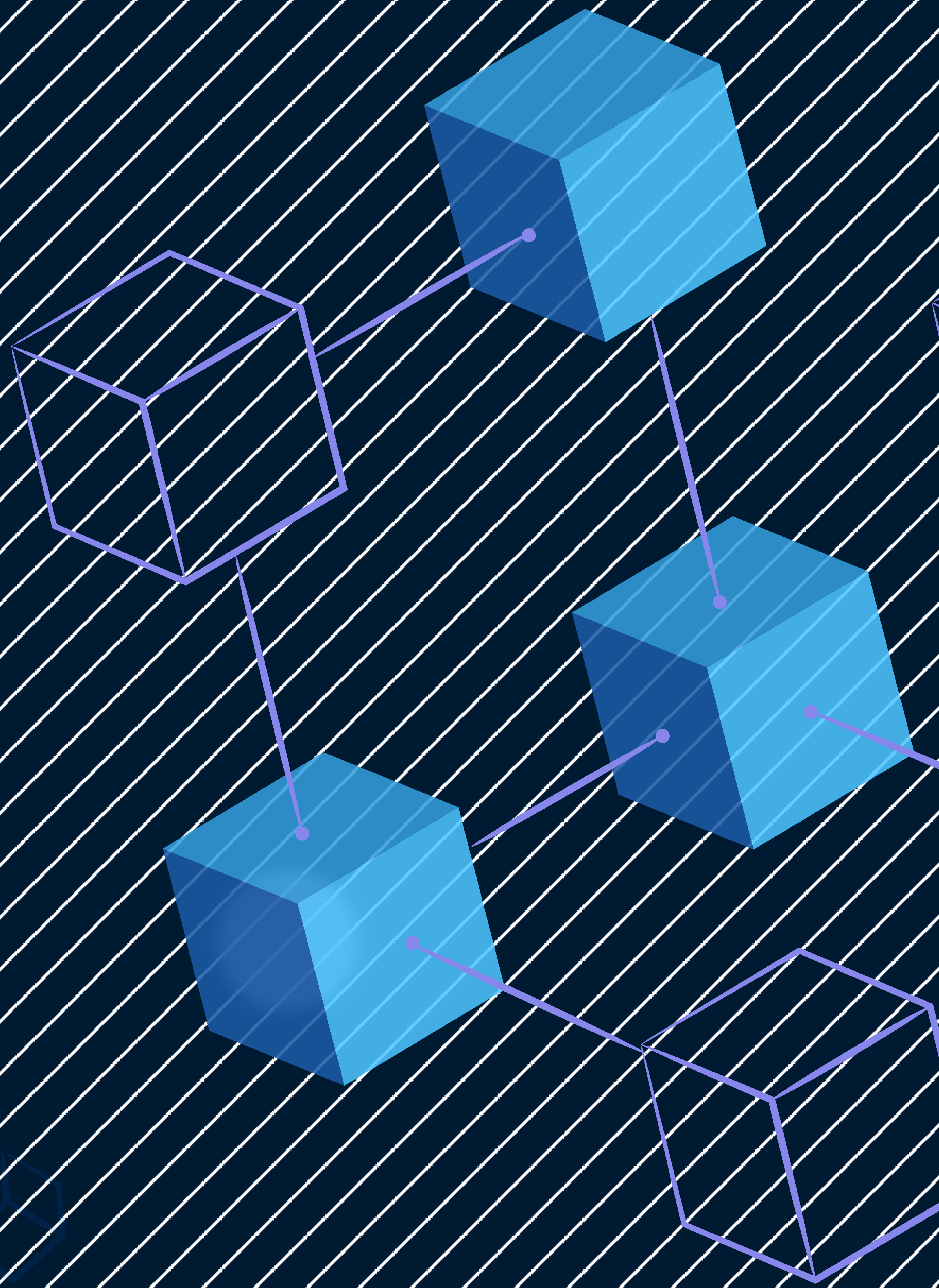


QuillAudits

# Audit Report January, 2022

For

Jigen



# Contents

Scope of Audit	01
Check Vulnerabilities	01
Techniques and Methods	02
Issue Categories	03
Number of security issues per severity.	03
Introduction	04
A. Contract - Jigen.sol	05
High Severity Issues	05
Medium Severity Issues	05
Low Severity Issues	05
Informational Issues	05
1. Some Variables can be made public	05
2. Typo	05
Functional Tests	06
Functionality Tests Performed	07
Automated Tests	09
Closing Summary	10



## Scope of the Audit

The scope of this audit was to analyze and document the Jigen 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 analysed 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.

### 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 the security of smart contracts.

### Code Review / Manual Analysis

Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the 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 the possibilities of optimization of code to reduce gas consumption.

### Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis, Theo.



## Issue Categories

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

Risk-level	Description
High	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 be fixed before moving to a live environment.
Medium	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	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 issues that 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	0	0	0
Closed	0	0	0	2

## Introduction

During the period of **December 20, 2021 to December 26 , 2021** - QuillAudits Team performed a security audit for the Jigen smart contract.

The code for the Jigen contract was obtained from:  
<https://github.com/jigenapp/jigen-contracts>

V	Date	Commit ID/Contract address	File
1	December	2e48f0de6df2cc392cbfa1224c3146e6cd797342	jigen.sol
2	December	9fe632586ab8c40dfe11a510facd3b4187a82f6	jigen.sol

The contract was deployed and tested on Ropsten and you can find it here:  
Jigen: [0xb5c6689e3f27a8c3b445d558be34d1a6e3582537](https://ropsten.etherscan.io/address/0xb5c6689e3f27a8c3b445d558be34d1a6e3582537)



## Issues Found – Code Review / Manual Testing

### A. Contract - Jigen.sol

#### High severity issues

No issues were found.

#### Medium severity issues

No issues were found.

#### Low severity issues

No issues were found.

#### Informational issues

##### 1. Some Variables can be made public

###### Description

The following variables can be made public instead of private:

- `_initialized`
- `_restrictionActive`
- `_tradingStart`
- `_maxTransferAmount`

###### Reason

The above variables can be made public so that when we need to update them it's convenient to check their current values.

Status: **Fixed**

##### 2. Typo

###### Description

The spelling of `_initialized` is mistyped as `_initlialized`.

Status: **Fixed**

## Functional Tests

Function Names	Testing results
transfer	Passed
transferFrom	Passed
burn	Passed
burnFrom	Passed
approve	Passed
transferOwnership	Passed
claimOwnership	Passed
increaseAllowance	Passed
decreaseAllowance	Passed
initAntibot	Passed
setTradingStart	Passed
setMaxTransferAmount	Passed
setRestrictionActive	Passed
whitelistAccount	Passed
unthrottleAccount	Passed



## Functionality Tests Performed

- Users should be able to transfer tokens not more than their balance. PASS
- Users should not be able to transfer if any of the conditions in the transactionThrottler are not met. PASS
- approve. PASS
- Users should be able to transferFrom tokens not more than their approval and also not more than the owner's(token owner not the contract owner) balance. PASS
- Users should not be able to transferFrom if any of the conditions in the transactionThrottler are not met. PASS
- Users should be able to burn tokens not more than their balance. PASS
- Users should be able to burnFrom tokens not more than their approval. PASS
- Only the current owner should be able to transferOwnership. PASS
- Only the nominated owner should be able to claimOwnership. PASS
- Users should be able to increaseAllowance. PASS
- Users should be able to decreaseAllowance. PASS
- initAntibot should only be called once and by the owner PASS
- Only the owner should be able to setTradingStart and \_tradingStart should be greater than the current block timestamp. PASS
- Only the owner should be able to setMaxTransferAmount. PASS

- Only the owner should be able to setRestrictionActive. PASS
- Only the owner should be able to whitelistAccount. PASS
- Only the owner should be able to unthrottleAccount. PASS





## Automated Tests

### Mythril

Mythril is a security analysis tool for EVM bytecode. It detects security vulnerabilities in smart contracts built for Ethereum, Hedera, Quorum, Vechain, Roostock, Tron and other EVM-compatible blockchains. It uses symbolic execution, SMT solving and taint analysis to detect a variety of security vulnerabilities. Mythril raised the following concerns:

```
The analysis was completed successfully. No issues were detected.
```

### Contract Library

Contract-library contains the most complete, high-level decompiled representation of all Ethereum smart contracts, with security analysis applied to these in real-time.

We performed analysis using the contract Library on the Ropsten address of the Jigen contract used during manual testing:

Jigen: [0x09dc564Ed0FD6B25cE6de35bf5C7a16576e5122F](#)

### Results

No major issues were found. All the other issues have been categorized above according to their level of severity.

## Closing Summary

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

The audit showed no issues.





## Disclaimer

Quillhash audit is not a security warranty, investment advice, or an endorsement of the Jigen 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 Jigen Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.





# Audit Report January, 2022

For

# Jigen



QuillAudits

📍 Canada, India, Singapore, United Kingdom

🌐 [audits.quillhash.com](https://audits.quillhash.com)

✉️ [audits@quillhash.com](mailto:audits@quillhash.com)