QuillAudits

Audit Report
March, 2021

OctoLend.

Contents

Scope of Audit	01
Techniques and Methods	01
Issue Categories	02
Introduction	04
Issues Found – Code Review/Manual Testing	04
Summary	07
Disclaimer	08

Scope of Audit

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

Check Vulnerabilities

- 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 contracts 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

	High	Medium	Low	Informational
Open	0	0	4	0
Closed	0	0	0	0

Introduction

During the period of March 18 2021 to March 20 2021 - Quillhash Team performed a security audit for OctoLend smart contracts.
The code for the audit was taken from the following BscScan Link where the contract is deployed:

https://www.bscscan.com/address/0x829958ccf6489c5b503c731cb2cbefe5d9bdc158#code

Issues Found – Code Review / Manual Testing

About the contract
A BEP-20 token contract with mintable and burnable features.

The contract also includes additional functionality of **whitelisting** and **blacklisting** any particular address. This function, however, is only accessible to the **owner** of the contract.

High severity issues

No Issues found.

Medium severity issues

No Issues found.

Low level severity issues

1. **Functions lack a Zero Address Check**
   Line no - 759,763

   Description:
   The contract includes some important functions that do not check whether or not the address passed as a parameter is a Zero Address. The following functions should implement a Zero Address Check

- **whiteList(address _address)**
- **blackList(address _address)**

Recommendation:
It is recommended to include a zero address check on the address parameters passed in any function.

2. **Unused Library Functions Found**
   Line no - 386,406

   Description:
   The SafeMath Library includes functions that are not used throughout the contract.

   While this affects gas optimization its also considered a bad practice to include unused functions or variables in the contract.

   Recommendation:
   Remove functions or variables that are not used in the contract.

3. **External visibility should be preferred**
   Line no - 99,108

   Description:
   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:
   - **renounceOwnership()**
   - **transferOwnership(address newOwner)**

   Recommendation:
   The above-mentioned functions should be assigned external visibility.

4. Coding Style Issues

Coding style issues influence code readability and in some cases may lead to bugs in future. Smart Contracts have a naming convention, indentation and code layout issues. It's recommended to use the Solidity Style Guide to fix all the issues.

Consider following the Solidity guidelines on formatting the code and commenting for all the files. It can improve the overall code quality and readability.

Closing Summary

Overall, smart contracts are well written and adhere to guidelines.
During the initial audit procedure, no high severity issues were found.
No instances of Re-entrancy or Backdoor Entry were found during the audit procedure.
Some imperative low severity issues as well as gas optimization issues, however, have been found and documented above.

Disclaimer

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

OctoLend.

QuillAudits

Canada, India, Singapore and United Kingdom

audits.quillhash.com

hello@quillhash.com