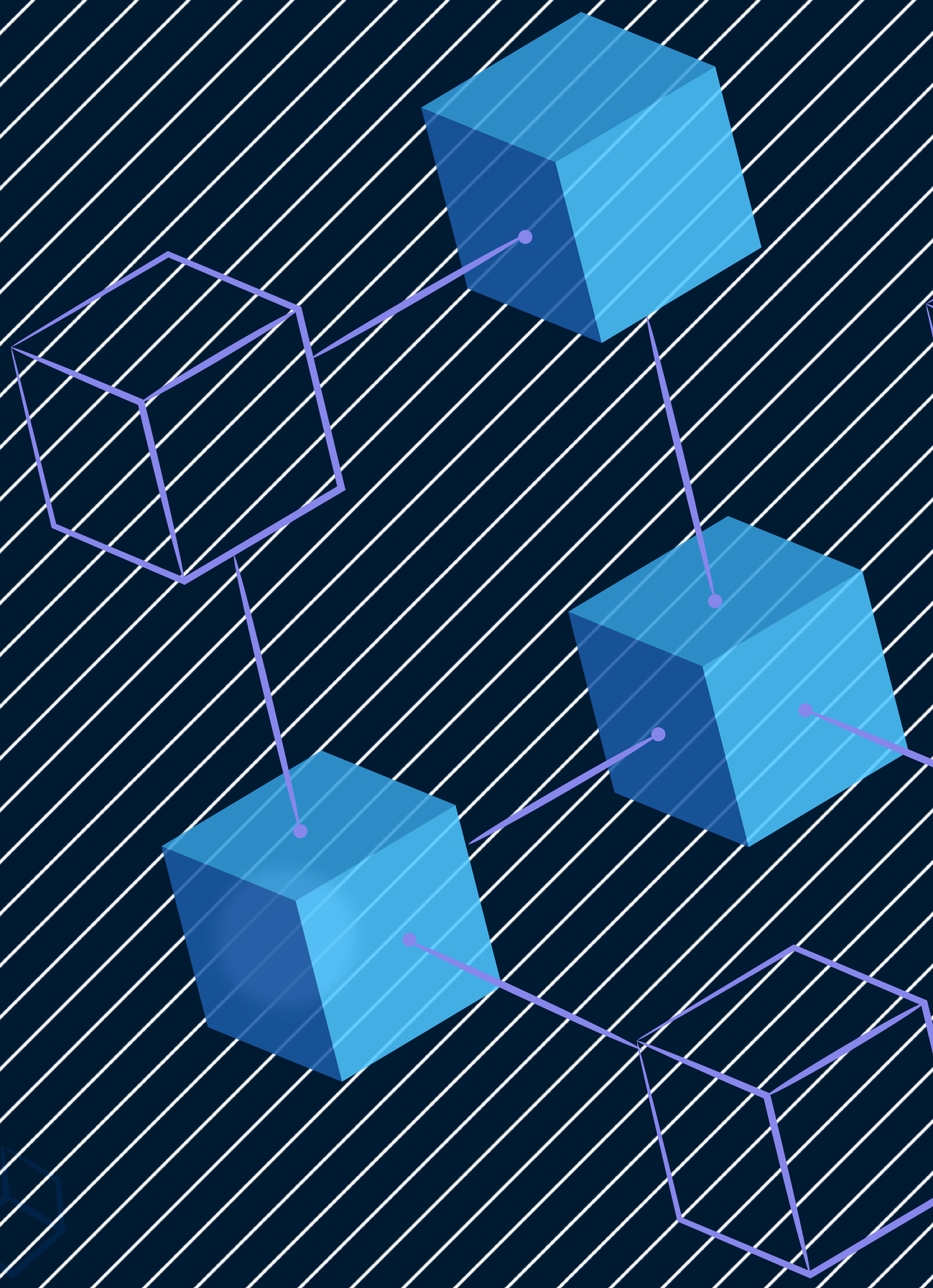
QuillAudits

Audit Report
October, 2021



For

YAN
the polygon launchpad

Contents

Scope of the Audit

The scope of this Audit was to analyze and document the YAAN 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
- BEP20 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 BEP20 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	2	1
Acknowledged	0	0	0	0
Closed	0	0	0	0

Introduction

During the **period of 11th October 2021 to 12th October 2021** - QuillAudits Team performed a security audit for **YAAN** smart contracts.

Contract list:

- yaan.sol

Note	Date	Commit hash
Version 1	11/10/2021	https://polygonscan.com/address/0xfe0a69a2fdb58e5beeecd07f78806c9dd0a54501

Issues Found

High severity issues

No issues were found.

Medium severity issues

No issues were found.

Low severity issues

1. **Range check while updating percent**

   Description
   In the setPercent() function, there is no check that ensures the _percent to stay within [0,100] range. An invalid percent value may cause unexpected behavior of the contract.

   Remediation
   We recommend adding a require statement that checks if (0<= _percent && _percent <= 100).

   Status: OPEN

2. **No check for address(0) before assignment:**

   Description
   In the following functions:
   - setAddressToChange()
   - setAddressToSend()

   There is no check to ensure that the incoming address in the parameters is not address(0).

   Remediation
   We recommend adding a require statement that checks if (incoming address != address(0)).

   Status: OPEN

Informational issues

3. Lack of event emissions

Description:
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:

- setPercent()
- setAddressToChange()
- setAddressToSend()
- changeOwnership()

Remediation:
We recommend emitting an event to log the updated variables.

Status: OPEN

Functional test

We did functional tests for different contracts manually. Below is the report:

Function Names	Technical Result	Logical Result
constructor	PASS	PASS
mint	PASS	PASS
burn	PASS	PASS
transfer	PASS	PASS
transferFrom	PASS	PASS
approve	PASS	PASS
setChangeStatus	PASS	PASS
setPercent	PASS	PASS
setAddressToChange	PASS	PASS
setAddressToSend	PASS	PASS
changeOwnership	PASS	PASS

Results

No major issues were found in all the test cases. All functions are working fine as per the logic and standard functions.

Closing Summary

Overall, smart contracts are very well written and adhere to guidelines.

No instances of Integer Overflow and Underflow vulnerabilities or Back-Door Entry were found in the contract, but relying on other contracts might cause Reentrancy Vulnerability.

Disclaimer

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

Audit Report
October, 2021

For

YAAN
the polygon launchpad

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com