Sheesha
FINANCE

SMART CONTRACT AUDIT

ZOKYO.

Mar 24, 2021 | v. 1.0

PASS

Zokyo's Security Team has concluded that this smart contract passes security qualifications to be listed on digital asset exchanges.

TECHNICAL SUMMARY

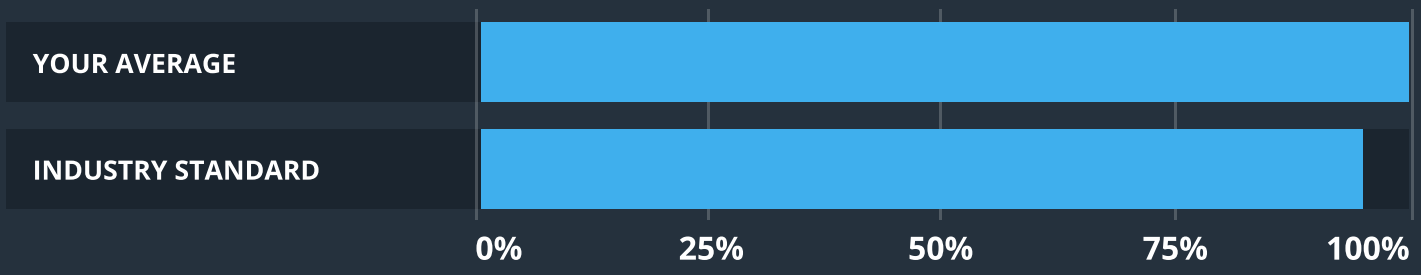
This document outlines the overall security of the Sheesha smart contracts, evaluated by Zokyo's Blockchain Security team.

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

Contract Status

There were no critical issues found during the audit.

Testable Code



Testable code is 100%, which is above the industry standard of 95%.

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that's able to withstand the Ethereum network's fast-paced and rapidly changing environment, we at Zokyo recommend that the Sheesha team put in place a bug bounty program to encourage further and active analysis of the smart contract.

TABLE OF CONTENTS

AUDITING STRATEGY AND TECHNIQUES APPLIED

The Smart contract's source code was taken from the following github repository – https://github.com/smartchain2030/sheesha-finance.
Our review focused on the commit hash – a8aef9e4d342abf5b6cb315de3849d93372a15d9.

Zokyo team audited LGE.sol contract only.

Requirements:

Development Outline:
https://docs.google.com/document/d/1wCaNLsvC3tPcNttH4b6vxNnf0KY2-fq8o7xv08249eI/edit?usp=sharing.

Throughout the review process, care was taken to ensure that the token contract:

- Implements and adheres to existing Token standards appropriately and effectively;
- Documentation and code comments match logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices in efficient use of gas, without unnecessary waste;
- Uses methods safe from reentrance attacks;
- Is not affected by the latest vulnerabilities;
- Whether the code meets best practices in code readability, etc.

Zokyo's Security Team has followed best practices and industry-standard techniques to verify the implementation of Sheesha smart contracts. To do so, the code is reviewed line-by-line by our smart contract developers, documenting any issues as they are discovered. Part of this work includes writing a unit test suite using the Truffle testing framework. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1	Due diligence in assessing the overall code quality of the codebase.
2	Cross-comparison with other, similar smart contracts by industry leaders.
3	Testing contract logic against common and uncommon attack vectors.
4	Thorough, manual review of the codebase, line-by-line.

SUMMARY

There were no critical issues found during the audit. All the mentioned findings may have an effect only in case of specific conditions performed by the contract owner.

Contracts are well written and structured. The findings during the audit have no impact on contract performance or security, so it is fully production-ready.

STRUCTURE AND ORGANIZATION OF DOCUMENT

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged “Resolved” or “Unresolved” depending on whether they have been fixed or addressed. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the ability of the contract to compile or operate in a significant way.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to operate in a way that doesn’t significantly hinder its behavior.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate.

COMPLETE ANALYSIS

HIGH | UNRESOLVED

Vesting logic applied to dev address is not part of the repository.

Recommendation:
In the scope of SHEESHA contract constructor execution, deploy new vesting contract and pass allocated dev funds to it.

HIGH | RESOLVED

Method transferVaultRewards & transferVaultLPewards from contract SHEESHA lets contract owner transfer rewards supply to vault multiple times.

Recommendation:
Decline method invocations if they already distribute executed.

HIGH | RESOLVED

Contract owner has the ability to transfer all ETH & tokens to itself.

Recommendation:
Add methods to pause all contract operations & to refund supplied ETH instead of transferring everything to the Contract owner address.

MEDIUM | RESOLVED

Method emergencyDrain24hAfterLiquidityGenerationEventIsDone of contract LGE is marked with modifier payable.

Recommendation:
Remove payable modifier, as the method does not utilize any passed ETH to it.

LOW | RESOLVED

Method `isUserEisting` from contract LGE has a typo.

Recommendation:
Rename method to `isUserExisting`.

INFORMATIONAL | UNRESOLVED

Smart contract is not covered by NatSpec annotations.

Recommendation:
Cover by NatSpec all Contract methods.

CODE COVERAGE AND TEST RESULTS FOR ALL FILES

Tests written by Zokyo Secured team

As part of our work assisting Sheesha in verifying the correctness of their contract code, our team was responsible for writing integration tests using the Truffle testing framework.

Tests were based on the functionality of the code, as well as a review of the Sheesha contract requirements for details about issuance amounts and how the system handles these.

Contract: LGE

- ✓ should set sheeshaGlobals correctly (64ms)
- ✓ should set contract start timestamp correctly (72ms)
- ✓ should getSecondsLeftInLiquidityGenerationEvent correctly (98ms)
- ✓ shouldn't getSecondsLeftInLiquidityGenerationEvent when event over (209ms)
- ✓ shouldn't createUniswapPairMainnet if it is already created (110ms)
- ✓ should addLiquidity correct (1224ms)
- ✓ shouldn't addLiquidity if liquidity generation event over (135ms)
- ✓ should addLiquidityToUniswapSHEESHAxWETHPair correctly (3751ms)
- ✓ shouldn't addLiquidityToUniswapSHEESHAxWETHPair if liquidity generation already finished (3309ms)
- ✓ should claimAndStakeLP correctly (7675ms)
- ✓ shouldn't claimAndStakeLP if there is nothing to claim (7643ms)
- ✓ shouldn't claimAndStakeLP when liquidity generation not finished yet (276ms)
- ✓ should emergencyDrain24hAfterLiquidityGenerationEventIsDone correctly (1136ms)
- ✓ shouldn't emergencyDrain24hAfterLiquidityGenerationEventIsDone when liquidity generation grace period still ongoing (174ms)

14 passing (2m)

FILE	% STMTS	% BRANCH	% FUNCS	% LINES	UNCOVERED LINES
contracts/	100.00	83.33	100.00	100.00	
LGE.sol	100.00	83.33	100.00	100.00	
All files	**100.00**	**83.33**	**100.00**	**100.00**	

We are grateful to have been given the opportunity to work with the Sheesha team.

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.

Zokyo's Security Team recommends that the Sheesha team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

ZOKYO.