Shapeshift

SMART CONTRACT AUDIT

ZOKYO.

June 23d, 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.

SCORE

99

TECHNICAL SUMMARY

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

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

Contract Status

LOW RISK

There were no critical and high issues found during the audit.

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 Shapeshift 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 3
Executive Summary 4
Structure and Organization of Document 5
Complete Analysis 6

AUDITING STRATEGY AND TECHNIQUES APPLIED

Staker smart contract's source code was taken from the archive provided by the Shapeshift's team. Archive's hash is given below:

SHA-256 (initial): d75eb1c8b12f892d753918f577a0f508cb8f0be7c9f053b0a86b7f218368db16
SHA-256 (latest): 56329f71fdb1c00990abe3c1acde1ea0a5bc4f1a32e5b8c1c3533d10656d20cf

Within the scope of this audit Zokyo auditors have reviewed the following contract(s):
RewardsDistributionRecipient.sol
StakingRewards.sol
StakingRewardsFactory.sol

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 Shapeshift 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.

EXECUTIVE SUMMARY

There were no critical or high issues found during the audit. Nevertheless, there were found several medium risk issues connected to Solidity version updates and OpenZeppelin version updates. All together the mentioned findings may have an effect only in case of specific conditions performed by the contract owner or are connected to the code style, extra variables and minor optimizations.

Nevertheless, all medium and low risk findings were successfully fixed by the Shapeshift team.

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 contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

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

MEDIUM | RESOLVED

Solidity version update

The solidity version should be updated. Throughout the project (including interfaces).
Issue is classified as Medium, because it is included to the list of standard smart contracts' vulnerabilities. Currently used version (0.5.16) is considered obsolete, which contradicts the standard checklist.

Recommendation:
You need to update the solidity version to the latest one in the branch - consider 0.7.6 or 0.8.4.

MEDIUM | RESOLVED

OpenZeppelin version update

OZ library version 2.3 should be updated.
Issue is classified as Medium, because it is included to the list of standard smart contracts' vulnerabilities. Currently used version (2.3.0) is considered obsolete, which contradicts the standard checklist. Library version should be updated together with the Solidity version - up to 3.4.0 or 4.1.0 depending on the chosen Solidity version.

Recommendation:
You need to update the OZ library version.

LOW | RESOLVED

Methods should be set as external

StakingRewardsFactory.deploy(address,uint256) (StakingRewardsFactory.sol#39-46)
StakingRewardsFactory.notifyRewardAmounts() (StakingRewardsFactory.sol#51-56)

Recommendation:
Declare methods as external.

INFORMATIONAL | RESOLVED

Variable should be declared as constant

StakingRewards.sol, line 23
Variable rewardsDuration should be declared as constant. It is set once and is never changed.

Recommendation:
Set the variable as constant.

INFORMATIONAL | RESOLVED

Rewards update

For now, rewards can be sent to the Staking pool only by the rewards distributor - the factory contract. Function StakingRewardsFactory.notifyRewardAmout() is designed in the way that rewards can be sent to the Staking contract only once (because allowed rewards amount is set to 0 and there is no interface to set it again). This makes staking pool one-time item. Consider adding the functionality to the Factory contract to update the allowed amount of rewards , so StakingRewardsFactory.notifyRewardAmout() can be used more times for the same pool.

Recommendation:
Verify the functionality works as intended.

We are grateful to have been given the opportunity to work with the Shapeshift 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 Shapeshift team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

ZOKYO.