BLOX STAKING Audit

Vesting and DEX Contracts

July 2021

By CoinFabrik

Introduction

CoinFabrik was asked to audit the contracts for the Vesting for Blox Staking. First we will provide a summary of our discoveries and then we will show the details of our findings.

Summary

The contracts audited are from the Github repository at https://github.com/bloxapp/ssv-network. The audit is based on the commit 95a79e44bbeb35b8c6a2bbd20e47762a4bb0dd11, fixes were applied and the commit ca99df216dc22d1e2170067b714c79d20aa003e0 was used to validate changes.

A new revision of minor changes was made in early August 2021 based on branch 'changes_after_review' (https://github.com/bloxapp/ssv-network/tree/changes_after_review) with commit `f1569ce9066963653539399174 77cd3e8a76d9c7` and no security issues were detected.

Contracts

The audited contracts are:

- `contracts/vesting/TokenVesting.sol`: Contract implementing vesting of ERC20 tokens.
- `contracts/DEX.sol`: Contract implementing exchange of CDTT for SSV tokens.
- `contracts/token/SSVToken.sol`: Token definition.

Analyses

The following analyses were performed:

- Misuse of the different call methods
- Integer overflow errors
- Division by zero errors
- Outdated version of Solidity compiler

- Front running attacks
- Reentrancy attacks
- Misuse of block timestamps
- Softlock denial of service attacks
- Functions with excessive gas cost
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Failure to use a withdrawal pattern
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures

Severity Classification

Security risks are classified as follows:

- **Critical:** These are issues that we manage to exploit. They compromise the system seriously. They must be fixed **immediately**.
- **Medium:** These are potentially exploitable issues. Even though we did not manage to exploit them or their impact is not clear, they might represent a security risk in the near future. We suggest fixing them **as soon as possible**.
- **Minor:** These issues represent problems that are relatively small or difficult to take advantage of but can be exploited in combination with other issues. These kinds of issues do not block deployments in production environments. They should be taken into account and be fixed **when possible**.
- **Enhancement:** These kinds of findings do not represent a security risk. They are best practices that we suggest to implement.

This classification is summarized in the following table:

SEVERITY	EXPLOITABLE	ROADBLOCK	TO BE FIXED
Critical	Yes	Yes	Immediately
Medium	In the near future	Yes	As soon as possible

Minor	Unlikely	No	Eventually
Enhancement	No	No	Eventually

Issues Found by Severity

Critical severity

CR-01 Denial Of Service in `vestings[]` struct

A malicious user could create an arbitrary amount of 'token vestings' for a specific user, all vesting 0 amount, thus growing the size of the vestings struct so that calls to any function looping through it would take a large amount of gas. As a result, these functions will fail. Examples include `totalVestingBalanceOf()`, `revokeAll()` and `withdrawFor()`.

Recommendation
Creating a vesting should require a positive amount, or even an amount bigger than a contract-defined threshold.

Status
The recommendation was followed and the issue has been fixed.

Medium severity

No issues found in this category.

Minor severity

MI-01 DEX contract allowing 0 rate

Failure to initialize DEX rate in 0 could lead to overflows and unexpected results.

Recommendation
Add a 'require' statement checking that the rate is nonzero.

Status
The recommendation was followed and the issue has been fixed.

Observations

During the audit, we noticed the SSV token was upgradeable (see, for example, link) and therefore its contract could be changed by the owner. We advised care with this, as the expected behaviour could be changed with any upgrade.

The contract was since updated and is no longer upgradeable.

Conclusion

We found the contracts to be simple, straightforward and with adequate documentation. A critical vulnerability was found allowing malicious users to impair the availability of certain functions, including `withdrawFor()`, a fix was proposed and applied. A minor-severity issue was found in the initialization of a token which was also fixed. No issues remain.

Disclaimer: This audit report is not a security warranty, investment advice, or an approval of the BLOX STAKING project since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee.