Katana Audit

September 2020

By CoinFabrik

Introduction

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

Summary

The contract audited is KatanaToken.sol, taken from the package Katana.rar, with md5 hash 4c1966ec0ccef5e8bf27d6cadbc8a8c7.

Contracts

The audited contract is:

- KatanaToken.sol: Katana Token, with delegation and checkpoint logic.

This contract uses OpenZeppelin's ERC20 and Ownable libraries.

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- Outdated version of Solidity compiler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

Detailed findings

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

Critical severity

No issues of critical severity have been found.

Medium severity

No issues of medium severity have been found.

Minor severity

Usage of now in delegateBySig

delegateBySig uses now instead of block number, which may be changed by the miner. This could be used to deny delegation in certain circumstances. However, it's unlikely this could be exploited. An improvement to this could be done by using the block number instead of block timestamp.

Enhancements

Uneven usage of safemath

The function getPriorVotes doesn't use safemath. While we didn't find that this would inevitably end in overflow, we advise using it.

Conclusion

We found the contract to be straightforward and have an adequate amount of documentation. No real issues were found, neither denial of service nor reentrancy attack in the token.

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