

HACKEN

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Customer: Samurai Legends
Date: March 1st, 2022

This document may contain confidential information about IT systems and the intellectual property of the Customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the Customer, or it can be disclosed publicly after all vulnerabilities are fixed – upon a decision of the Customer.

Document

Name	Smart Contract Code Review and Security Analysis Report for Samurai Legends.
Approved by	Andrew Matiukhin CTO Hacken OU
Type	BEP20 token; Staking
Platform	Binance Smart Chain / Solidity
Methods	Architecture Review, Functional Testing, Computer-Aided Verification, Manual Review
Repository	https://github.com/Samurai-Legends/contract_samurai-legends-tokens
Commit	aff9f451dff6854f29375ac21f8cedba8ee1101
Deployed contract	–
Technical Documentation	YES
JS tests	YES
Website	
Timeline	26 JANUARY 2022 – 28 FEBRUARY 2022
Changelog	01 FEBRUARY 2022 – INITIAL AUDIT 15 FEBRUARY 2022 – SECOND REVIEW 01 MARCH 2022 – THIRD REVIEW



Table of contents

Introduction	4
Scope	4
Executive Summary	5
Severity Definitions	7
Audit overview	8
Conclusion	8
Disclaimers	10

Introduction

Hacken OÜ (Consultant) was contracted by Samurai Legends (Customer) to conduct a Smart Contract Code Review and Security Analysis. This report presents the findings of the security assessment of the Customer's smart contract and its code review conducted between January 26th, 2022 - February 15th, 2022.

Scope

The scope of the project is smart contracts in the repository:

Repository:

https://github.com/Samurai-Legends/contract_samurai-legends-tokens

Commit:

[aff9f451dffc6854f29375ac21f8cedba8ee1101](https://github.com/Samurai-Legends/contract_samurai-legends-tokens/commit/aff9f451dffc6854f29375ac21f8cedba8ee1101)

Technical Documentation: Yes

- token:
https://docs.google.com/document/d/1xWn-6_e7ikK0ztd8irlur3swIMywQyfEXyRmzIhExQ8/edit
- staking:
https://docs.google.com/document/d/1FeXPZlqH3FQRCNAUV4DtBg-rMqLC1yEfEtqD__7r5KE/edit#

JS tests: Yes (included in the “test” directory)

Contracts:

[Array.sol](#)
[ERC20WithFees.sol](#)
[Generatable.sol](#)
[Koku.sol](#)
[Recoverable.sol](#)
[SamuraiLegends.sol](#)
[SamuraiLegendsStaking.sol](#)



We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the commonly known vulnerabilities that are considered:

Category	Check Item
Code review	<ul style="list-style-type: none">▪ Reentrancy▪ Ownership Takeover▪ Timestamp Dependence▪ Gas Limit and Loops▪ DoS with (Unexpected) Throw▪ DoS with Block Gas Limit▪ Transaction-Ordering Dependence▪ Style guide violation▪ Costly Loop▪ ERC20 API violation▪ Unchecked external call▪ Unchecked math▪ Unsafe type inference▪ Implicit visibility level▪ Deployment Consistency▪ Repository Consistency▪ Data Consistency
Functional review	<ul style="list-style-type: none">▪ Business Logics Review▪ Functionality Checks▪ Access Control & Authorization▪ Escrow manipulation▪ Token Supply manipulation▪ Assets integrity▪ User Balances manipulation▪ Data Consistency manipulation▪ Kill-Switch Mechanism▪ Operation Trails & Event Generation



Executive Summary

According to the assessment, the Customer's smart contracts are secured.



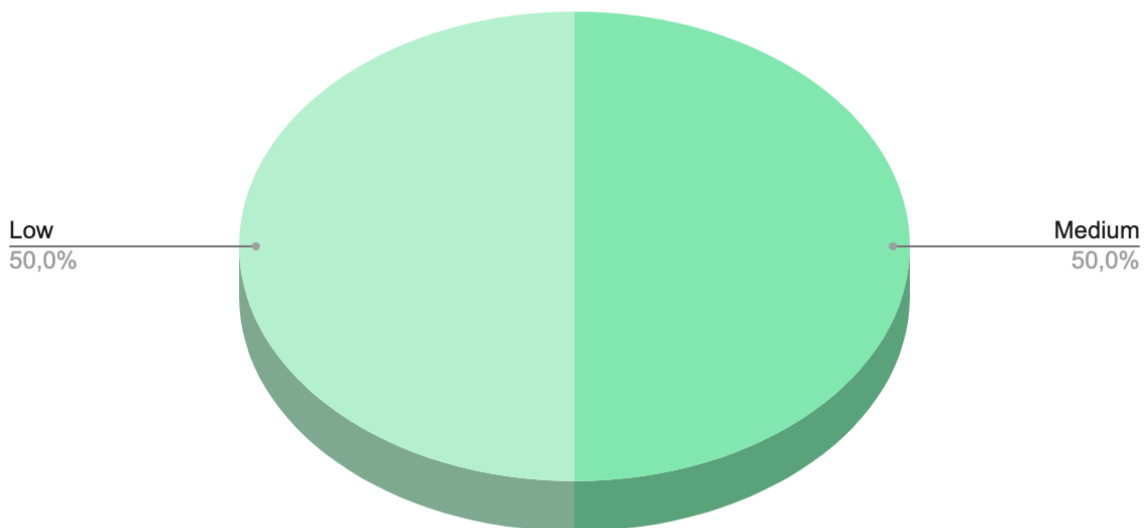
Our team performed an analysis of code functionality, manual audit, and automated checks with Mythril and Slither. All issues found during automated analysis were manually reviewed, and important vulnerabilities are presented in the Audit overview section. All found issues can be found in the Audit overview section.

As a result of the audit, security engineers found **1** medium and **1** low severity issue.

After the second review security engineers found that there were some require-statements removed from the code and tests were slightly updated. However, the code is still poorly covered by tests, and part of the tests are failing. So, for now, we still see **1** medium and **1** low severity issue.

After the third review security engineers found that code coverage is better now but still there are **1** medium and **1** low severity issue.

Graph 1. The distribution of vulnerabilities after the audit.



Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution

Audit overview

■■■■ Critical

No critical issues were found.

■■■ High

No high issues were found.

■■ Medium

1. Too low test coverage.

The test coverage is too low. For example, “SamuraiLegendsStaking” is covered only for about 56% of code branches, which is very low. “Koku” is covered only for 69%.

Scope: tests

Recommendation: Please improve code coverage to be at least 95% for statements and up to 100% for branches.

Status: Tests coverage increased in total 87% for statements and 76% for branches.

■ Low

1. Block timestamp.

Dangerous usage of `block.timestamp`. `block.timestamp` can be manipulated by malicious miners within 15 minutes.

Contracts: SamuraiLegendsStaking.sol

Functions: userClaimablePendingPercentage, lastTimeRewardActiveAt, rewardPerToken, withdraw, withdrawAll, claim, addReward, decreaseReward, resetReward, updateRewardDuration

Recommendation: Please consider relating on the `block.number` instead,

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

As a result of the audit, security engineers found **1** medium and **1** low severity issue.

After the second review security engineers found that there were some require-statements removed from the code and tests were slightly updated. However, the code is still poorly covered by tests, and part of the tests are failing. So, for now, we still see **1** medium and **1** low severity issue.

After the third review security engineers found that code coverage is better now but still there are **1** medium and **1** low severity issue.



Disclaimers

Hacken Disclaimer

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions).

The audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only – we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit can't guarantee the explicit security of the audited smart contracts.