HyruleSwap

Smart Contract Security Audit

March, 2021

TechRate

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by HyruleSwap team to perform an audit of smart contracts:

- *https://bscscan.com/address/0x7b0409a3a3f79baa284035d48e1dfd581d7d7654#code*
- *https://bscscan.com/address/0x922e11dd7f05d24dbb5397f8eeeb7bfd6e01f43f#code*
- *https://bscscan.com/address/0x76bd7145b99fdf84064a082bf86a33198c6e9d09#code*

The purpose of the audit was to achieve thefollowing:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Issues Checking Status Summary

№	Issue description.	Checking status
1	Compiler warnings.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed

19	Cross-function race conditions.	Passed
20	Safe Zeppelin module.	Passed
21	Fallback function security.	Passed

Compiler warnings

We compiled the three contracts with the solidity 0.6.12 compiler and we did not find any warning.

Conclusion: **No deviation noted**

Race conditions and Reentrancy. Cross-function race conditions

By analyzing the contracts, we did not find any bugs that arose from the nondeterministic timing of the execution. Moreover, we did not find any call to an external code before the end of the internal treatment.

Conclusion: **No deviation noted**

Possible delays in data delivery

We did not find any poorly optimized functions in their execution, which could create an abnormal delay in data delivery.

Conclusion: **No deviation noted**

Oracle calls

No oracle calls needed by the contracts, not applicable.

Conclusion: **No deviation noted**

Front running

HyruleSwap platform does not allow swapping currencies with their own contract. The app redirects to PancakeSwap, which allows users to set their own slippage. Not applicable.

Conclusion: **No deviation noted**

Timestamp dependence

No major security issues were found in the functions handling timestamp, no critical dependency.

Conclusion: **No deviation noted**

Timestamp dependence

No major security issues were found in the functions handling timestamp, no critical dependency.

Conclusion: **No deviation noted**

Integer Overflow and Underflow

No incrementing or decrementing variables, which could lead to an Integer Overflow and Underflow security issue.

Conclusion: **No deviation noted**

DoS with Revert

In the three contracts, there is no function that depends on a transaction between two addresses, which could lead to a DoS with a Revert risk.

Conclusion: **No deviation noted**

DoS with block gas limit

There are no transactions to an address array, which could lead to a DoS with a gas block limit.

Conclusion: **No deviation noted**

Methods execution permissions

The timelock contract functions are appropriately configured to prevent any risk leading to a wrong segregation of administration rights.

Conclusion: **No deviation noted**

Economy model

The HyruleSwap application logic is based on a correct economic model, the application functions correctly and the participants do not incur any financial losses. The rewards' distribution system is fair among Staking and Farming offers.

Conclusion: **No deviation noted**

The impact of the exchange rate on the logic

There is no significant impact of the exchange rate on the application logic.

Conclusion: **No deviation noted**

Private user data leaks

There is no significant risk of a private user data leaks.

Conclusion: **No deviation noted**

Malicious Event log

There is no significant risk for end users or HyruleSwap team related to a malicious event log.

Conclusion: **No deviation noted**

Scoping and Declarations

By analyzing the three contracts, we did not find any ineffective declaration of variable, which could lead to an improper scoping and declaration issue.

Conclusion: **No deviation noted**

Uninitialized storage pointers

There is no risk of uninitialized storage pointers thanks to the use of an in-memory array and appropriate variables declarations.

Conclusion: **No deviation noted**

Arithmetic accuracy

By analyzing three smart contracts, we did not find any Arithmetic accuracy issues, which could lead to the malfunction of the application.

Conclusion: **No deviation noted**

Design Logic

The Design Logic of the application is appropriate to allow its good functioning and its good maintainability.

Conclusion: **No deviation noted**

Cross-function race conditions

By analyzing the three smart contracts, we can evaluate that public functions are protected against Cross-function race attacks.

Conclusion: **No deviation noted**

Safe Zeppelin module

HyruleSwap’s contract applications use OpenZeppelin Contracts to help minimize the risk by using battle-tested libraries of smart contracts for Ethereum and other blockchains. It includes the most used implementations of ERC standards.

Conclusion: **No deviation noted**

Fallback function security

By analyzing the three smart contracts, we did not find any Fallback function security issues, which could lead to a malfunction of the application.

Conclusion: **No deviation noted**

Security Issues

High Severity Issues

No high severity issues found

Medium Severity Issues

No medium severity issues found

Low Severity Issues and Owner privileges

No low severity issues found

Conclusion

The various tests, that have been carried out to evaluate the risks associated with HyruleSwap smart contracts, allow us to conclude that:

Smart contracts do not contain any high severity issues!