SMC AUDITORS

Spherium Smart Contracts Audit

SMCAuditors

https://smcauditors.com / info@smcauditors.com

29th of August 2021

This document is the audit of Spherium smart contracts performed by SMCAuditors.

1. Executive Summary

This report was written to provide a security audit for Spherium (https://spherium.finance) smart contract. Spherium is a mobile first wallet which optimizes return on crypto assets. SMCAuditors conducted the audit focusing on whether Spherium smart contract is designed and implemented without any security vulnerabilities. The contract is listed below with its link.

- SpheriumRouter01.sol
  - https://gitlab.com/spherium/uniswap/periphery/-/blob/master/contracts/SpheriumRouter01.sol
- SpheriumMigrator.sol
  - https://gitlab.com/spherium/uniswap/periphery/-/blob/master/contracts/SpheriumMigrator.sol
- SpheriumFactory.sol

    - https://gitlab.com/spherium/uniswap/periphery/-/blob/master/contracts/SpheriumFactory.sol

We have run extensive static analysis of the codebase as well as standard security assessment utilising industry approved tools. There are no high level issues with the currently deployed contracts. Our findings are available in the next section.

2. Audit Findings

SpheriumRouter1.sol

Low Level Findings

1. Consider using the latest solidity version
2. Define “uint” explicitly in the code to avoid overflow and underflow issues. You can use “uint256” in general.
3. Following functions can be called by *external* modifier rather than *public*.
    a. *quote(uint256,uint256,uint256)*
    b. *getAmountOut(uint256,uint256,uint256)*
    c. *getAmountIn(uint256,uint256,uint256)]*
4. Line 321: Use local variables instead of state variables like *length* in a loop to reduce the gas consumption.

SpheriumMigrator.sol

Low Level Findings

1. Consider using the latest solidity version
2. Define “uint” explicitly in the code to avoid overflow and underflow issues. You can use “uint256” in general.

SpheriumFactory.sol

Low Level Findings

1. Consider using the latest solidity version

2. Define “uint” explicitly in the code to avoid overflow and underflow issues. You can use “uint256” in general.

Medium Level Findings

1. Reentrancy Risk on *createPair* function*.* Use Open Zeppelin ReentrancyGuard when calling external functions and apply checks-effects-interactions.

3. Conclusion

In this audit, we thoroughly analyzed the Spherium smart contracts. Overall, the smart contracts were well written and common security standards were used by the team. Our identified issues were informed and resolved by the team.

4. Disclaimer

This report is not advice on investment, nor does it guarantee adequacy of a business model and/or a bug-free code. This report should be used only to discuss known technical problems. It will be necessary to resolve addressed issues and conduct thorough tests to ensure the safety of the smart contract.