ETHOS
AUDIT

# Ocean Stake

Audit Report

Sep. 12, 2022

# Contents

# Executive Summary

## Audit Details

| | |
|---|---|
| Project Name | OceanStake |
| Codebase | https://bscscan.com/address/0x3fb39de0e9e344733723a7f7358132bdc0cf33d0#code |
| Initial Audit Date | September 12, 2022 |
| Revision Dates | - |
| Methodology | Manual |

## Methodology

This audit's objectives are to evaluate:

- Security-related issues
- Code quality
- Relevant documentation
- Adherence to specifications
- Adherence to best practices

This audit examines the possibility of issues existing along the following vectors (but not limited to):

- Single & Cross-Function Reentrancy
- Front Running (Transaction Order Dependence)
- Timestamp dependence
- Integer Overflow and Underflow
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Number rounding errors
- DoS with (Unexpected) Revert
- DoS with Block Gas Limit
- Insufficient gas griefing
- Forcibly sending native currency
- Logical oversights
- Access control
- Centralization of power
- Logic-Specification Contradiction
- Functionality duplication
- Malicious token minting

The code review conducted for this audit follows the following structure:

1. Review of specifications, documentation to assess smart contract functionality
2. Manual, line-by-line review of code
3. Code's adherence to functionality as presented by documentation
4. Automated tool-driven review of smart contract functionality
5. Assess adherence to best practices
6. Provide actionable recommendations

## Contract Details

|   |   |
|---|---|
| Contract ID | 0x3fB39dE0E9E344733723A7F7358132bdc0cf33D0 |
| Network | BSC |
| Language | Solidity |
| Compiler | v0.8.16+commit.07a7930e |
| Verification Date | - |

## Result Summary

Ethos' audit of the **Ocean Stake** smart contract has concluded with a **PASSING** result, meaning the contract is mostly safe from external threats. The initial review identified several low risk and informational items that have been acknowledged by the team. **None of the issues require a redeployment of the smart contract since the issues don't represent any significant risk to the value locked within the contract**. The remaining report includes all issues identified in the initial review, as well as the revised status post resolution by the team where applicable.

- The smart contract is BNB rewards contract on the Binance Smart Chain, containing several advancements and modifications from the miner meta

- It allows users to deposit BNB into the contract which are time-locked on deposit and redistributed to users over time

- The rate of redistributions depends on the length of time that funds are kept in the contract and varies based on the rate of increase of total value locked, 5% daily rewards max (%1 base + %2 hold bonus + %2 balance bonus), Max 200% rewards payout

- There are various levels of referral bonuses up to 8% (1st level 4%, 2nd level 2%, 3rd level 1%, 4th level 1%)

- There is a 10% fee applied to deposits, no fees on compounding or withdrawals

- The contract cannot be closed or shut off at any point after deployment

To conclude, this smart contract does what it is designed to and does not contain any backdoors or owner privileges allowing the contract owner to unexpectedly stop or drain the contract of its locked value.

## Issues Reported

| Severity | Unresolved | Acknowledged | Resolved |
|---|---|---|---|
| High | 0 | 0 | 0 |
| Medium | 0 | 0 | 0 |
| Low | 0 | 0 | 0 |
| Info | 6 | 0 | 0 |

## Issues Summary

| ID | Title | Severity | Status |
|---|---|---|---|
| OS-0 | Block timestamp reliance | Info | Acknowledged |
| OS-1 | Division before multiplication | Info | Acknowledged |
| OS-2 | Strict equality | Info | Acknowledged |
| OS-3 | Low level calls | Info | Acknowledged |
| OS-4 | Public functions could be declared external | Info | Acknowledged |
| OS-5 | Costly operations in loop | Info | Acknowledged |

# Detailed Findings

## Code Documentation

The code contains sufficient commenting.

## Adherence to Specifications

The Ocean Stake smart contract adheres to the smart contract functionality described by the project documentation and is in line with its intended usage.

## Adherence to Best Practices

The Ocean Stake smart contract adheres to the best practices associated with a standard EVM compatible Solidity smart contract.

## OS-0 – Block Timestamp reliance

**Severity**: Informational **Status**: Acknowledged

**Description**: Functions use statements that rely on a block timestamp comparison.

**Risk:** Miners can manipulate block.timestamp value to exploit the require statement and contract.

**Recommendation**: Avoid using block.timestamp for comparison logic.

## OS-1 – Division before multiplication

**Severity**: Informational **Status**: Acknowledged

**Description**: Solidity integer division might truncate. As a result, performing multiplication before division can sometimes avoid loss of precision.

**Risk:** In general, it's usually a good idea to re-arrange arithmetic to perform multiplication before division, unless the limit of a smaller type makes this dangerous.

**Recommendation**: Consider ordering multiplication before division.

## OS-2 – Strict equality

**Severity**: Informational **Status**: Acknowledged

**Description**: Use of strict equalities that can be easily manipulated by an attacker.

**Risk**: Some comparisons might return always false if the value being compared is a very small remainder close to 0 or some integer, causing the comparison to always fail.

**Recommendation**: Avoid the use of strict equality to determine if an account has enough value or tokens.

## OS-3 – Low-level calls

| | |
|---|---|
| **Severity**: Informational | **Status**: Acknowledged |

**Description**: The use of low-level calls is error-prone.

**Risk**: Low-level calls do not check for code existence or call success.

**Recommendation**: Avoid low-level calls. Check the call success. If the call is meant for a contract, check for code existence.

## OS-4 – Public functions that could be declared external

| | |
|---|---|
| **Severity**: Informational | **Status**: Acknowledged |

**Description**: Public functions that are never called by the contract should be declared external to save gas.

**Risk**: Gas optimization

**Recommendation**: Use the external attribute for functions never called from the contract.

## OS-5 – Costly operations inside loop

| | |
|---|---|
| **Severity**: Informational | **Status**: Acknowledged |

**Description**: Costly operations inside a loop might waste gas, so optimizations are justified.

**Risk**: Gas optimization

**Recommendation**: Use a local variable to hold the loop computation result.