safepress

Smart Contract Audit

For smart contract vulnerabilities,
security exploits and attack vectors

Code Review And Security Report

Important: This document likely contains critical information about the Client's software and hardware systems, security susceptibilities, descriptions of possible exploits and attack vectors. The document shall remain undisclosed until any significant vulnerabilities are remedied.

CLIENT:	Covalent
START DATE:	7th March 2022
END DATE:	14th March 2022
TYPE, SUBTYPE:	Staking

Scope

Repository:	https://github.com/covalenthq/covalent-operational-staking-audit
Commit hash:	259af726b006fb2aba7dbba7968603279bb743d5
Documentation:	No documentation apart from inline is available
Tests:	Passing
Auditors:	Dima, Sumit
Review & Approval:	Ruby
Smart Contract Audited:	OperationalStaking.sol

Definitions of vulnerability classification

Severity	Definition
Critical	Bug / Logic failures in the code that cause loss of assets / data manipulation.
High	Difficult to exploit problems which could result in elevated privileges, data loss etc.
Medium	Bug / Logic failures in the code which need to be fixed but cannot lead to loss of assets / data manipulation.
Low	Mostly related to unused code, style guide violations, code snippets with low effect etc.

The smart contracts were found with the following vulnerabilities:

High-level Vulnerabilities

- [ ] Ownership Takeover

Informational

- [ ] Solidity Style Guide violations - multiple occasions.

Executive Summary

Based on the audit findings the Client's contracts are: Secured

Not Secure	Insufficiently Secured	Secured	Well Secured

Found Issues / Vulnerabilities:

Function initialize

After deployment the function can be called by ANY account thus taking over the ownership of the contract.
Recommendation (alternatives):

1. Call this function from a deployment script immediately after deploying the smart contract
2. Hardcode the address that can call this function after deployment and add a corresponding check or a custom modifier

Solidity Style Guide violations:

1. The contract is targeting Solidity v0.8.4 while v0.8.12 is available. There were multiple bug- and vulnerability fixes between the versions, thus it makes sense to stick to the latest stable version. Version 0.8.13 is currently experimental

2. The public constant DIVIDER is assigned 10**18 while it should be written as “1 ether”

3. The public constant rewardRedeemThreshold is written in camel case, while constants must be capitalized & snake cased. Besides, the constant is assigned a value of “10*8” while scientific notation is recommended in such cases: “1e8”

4. There are multiple occasions of unreadable lines with line lengths exceeding 120 characters. All such cases should be split into several lines:
   L# 47, 52, 62, 72, 319, 365, 382, 383, 387, 393, 404

5. The contract storage variables in lines L#15-23 require explicit access modifier “public”

6. Events declared in lines L#47-64 must be separated with empty lines for readability

7. Some functions use inconsistent indentation (2 spaces instead of 4), ex. L# 113-114, and everywhere inside if-else blocks

8. All the functions with the “external” modifier starting from L# 201 should be placed before functions with the “internal” modifier starting from L# 178