Canonical LLC

Security Assessment

# SingularityNet

# Unbonded Staking Contracts

# Executive Summary

This verification report is for the SingularityNET Unbonded staking system. The staking system comprises multiple contracts that work together to allow users to stake tokens into a pool and receive rewards based on the duration they keep their tokens staked.

The pool and its rewards are managed by a trusted administrator. This administrator, or any entity that gains control of the administrator key, holds custody of the tokens in the pool. The administrator has the authority to remove all the tokens at their discretion or to make other modifications necessary for reward distribution.

Reward distribution is at the discretion of the administrator. However, users can opt to withdraw their tokens from the pool, provided there are sufficient tokens remaining in the pool.

The contracts aim to ensure that only the user and the administrator have access to the tokens locked in a pool.

This security assessment is the second audit that was conducted. In the first audit, issues were identified and subsequently addressed. In the time since, another vulnerability was discovered.

Through a combination of manual review and white box testing, no additional security issues were identified.

The focus of the audit was to uncover security vulnerabilities, with special attention given to potential adversarial use of the contract.

No security vulnerabilities were found. The audit contains informational issues that could be addressed to improve the maintainability of the codebase.

# Issue Summary

No issues requiring action were found. Below are informational issues we believe could be addressed to improve maintenance of the code base.

| ID | Title | Severity | Status |
|---|---|---|---|
| 1 | Duplicated Hard Coded Strings | Informational | Unresolved |
| 2 | Boolean Blindness | Informational | Unresolved |
| 3 | Unnecessary Monadic Action | Informational | Unresolved |
| 4 | Logging Inconsistency | Informational | Unresolved |
| 5 | Missing Function Name in Logs | Informational | Unresolved |

# Information

| | |
|---|---|
| Repo | https://github.com/mlabs-haskell/singularitynet-onchain |
| Commit | 9631f3e1f5aa6962a08eb2116d1be0e1ad90d9d3 |
| Address | Parametric. Depends on the parameters used. |

# 1. Duplicated Hardcoded Strings

| Severity | Status |
| --- | --- |
| Informational | Unresolved |

## Description

In the files PType.hs the definitions for `PEntryHRec` and `PEntryHRec` include duplicated type level strings. This duplication poses a small maintance hazard. The code duplication could be reduced the with type aliases.

The same issue is present `PUnbondedPoolParamsHRec` and `PUnbondedPoolParamsFields`

## Recommendation

Use a type alias to reduce duplication.

# 2. Boolean Blindness

| Severity | Status |
| --- | --- |
| Informational | Unresolved |

## Description

In the file Types.hs the `StateDatum` constructor includes a `Bool` represent whether the pool is open or closed. This could lead to “Boolean Blindness.” Instead use an enumeration for `OpenState = Open | Closed` to make it clearer what this field means.

## Recommendation

Use a custom type instead of a `Bool`

## 2. Unnecessary Monadic Action

| Severity | Status |
| --- | --- |
| Informational | Unresolved |

### Description

In the file UnbondedPool.hs on line 637 there is the following line:

`(\txOut -> pure $ pfield @"value" # txOut) <=<`

The action in not inherently monadic and could be fmapped instead.

### Recommendation

fmap function instead promoting to a monadic action.

# 2. Logging Inconsistency

| Severity | Status |
| --- | --- |
| Informational | Unresolved |

## Description

In the file Utils.hs the functions `getTokenCount`, `foldCsMap`, `foldTnMap` and `getOutputSignedBy` follow a different convention for logging errors. Where as all other functions use the format `FUNCTION_NAME: ` these function use the format `(FUNCTION_NAME)`.

## Recommendation

Use a consistent format for all logging operations.

## 2. Missing Function Name In Logs

| Severity | Status |
| --- | --- |
| Informational | Unresolved |

### Description

Almost all the log messages include the function name as a prefix. However, this convention was dropped in a few places. Specifically, the functions `evalCs` and `evalTnAndAmount` log messages' "predicate on CurrencySymbol not satisfied" and "predicate on TokenName/amount not satisfied" respectively are missing their function names.

### Recommendation

Add the “FUNCTION_NAME: “ prefix to the log messages for `evalCs` and `evalTnAndAmount`.

# Disclaimer

This report is not an endorsement of SingularityNET or the smart contract being audited. The report should not be used to make investments or other financial decisions.

Using the Cardano blockchain entails risk, and the user of any smart contract must perform their own due diligence.

This report should not be misconstrued as a guarantee of the security properties of the smart contract.