Enigma Ventures S.L.
info@enigmaventures.xyz

Stakingverse - Liquid staking token

8 January 2025 - v2

OVERVIEW

Stakingverse is launching a liquid staking token (LST) solution to enable users of an existing vault to represent their share in the form of a transferable token. Users who stake in a vault receive shares to represent their increasing in value (LYX - Lukso token) stake - share of the vault. By adding LST users can transfer their shares (mint) LST or directly receive LST by sending LYX to LST contract.

GOALS

1. Asses integration of LST with Vault contract.
2. Evaluate security and provide recommendations for LST and Vault.

NON-GOALS

1. Guarantee full security of the code and its changes after this assessment.
2. Misplacement or loss of keys and control over deployed contracts.

SPECIFICATIONS

Files under review:

File name	Commit hash	Revised diff
LiquidStakingToken.sol -> SLYXGToken.sol	b85fa1d5e9a759dca18f844cfab972adc34c7c91	0c4c613b4459a5431f8767470e2a56f6bd69f7144baad8e69eeda581d6fabe35
LiquidStakingTokenAutoMintExtension.sol	b85fa1d5e9a759dca18f844cfab972adc34c7c91	

ASSESSMENT

Critical: LiquidStakingToken assumes shares is the amount of LST tokens

Revision: fixed

LiquidStakingToken assumes shares LST contract holds are the same amount of tokens LST contract minted. There are no assertions to enforce such assumptions currently in the code.

This assumption fails in a case of rewards being issued to a vault. After rewards are accumulated, shares cost rises relative to LYX (not LST). In such a case, the amount of LYX being minted will be higher than the amount of shares representing this LYX amount. This results in the amount of LST being higher than the amount of shares owned by the LST contract.

Evaluation and prove

The shares amount is used in getNativeTokenValue to calculate how much of LST needs to be burned to retrieve represented LYX (staked LYX). This is used by _afterTokenTransfer in a case of burn (to == address(0)). Please see the following gist for a failing test and changes required to fix the test to properly account for LYX amount. To run the test: *forge test --match-contract FailureCase -vv*.

Recommendation

The recommendation to address the issue comes down to removal of vault.sharesOf usage from LST and using directly totalSupply of LST itself, as it is a source of truth for the amount of LST minted. As a reference please see the following gist for rough changes (diff patch) may be needed to address the issue.

Medium: LiquidStakingTokenAutoMintExtension is not required

Revision: fixed

Addition of extension is not required. LST contract itself is capable of hosting the same direct deposit functionality. Having additional contracts brings more complexity to the system for no obvious benefit.

Additionally, the first staker vault may reserve a minor amount of wei to prevent share inflation issue. Thus, logic of deposit within extension must not use msg.value through transfer of a stake. If rounder error or first staker cases are present, deposit will revert and result in inability to make deposits.

Recommendation

LST contract could allow direct deposit to its associated vault and transfer of a stake within a single method, similar as it is implemented in the extension code. Additionally LST could accept raw value transfers via *receive() payable* and consider them as deposits instead of reverting as LSP7 default behavior.

When it comes to proper value accounting within the deposit function, it should be calculated using the difference of a balance increase (before/after deposit to vault). This will account for scenarios such as: first staker and minor wei reservation, and 1 wei rounding error.

Recommendation: Token transfer pause control

Revision: N/A

LiquidStakingToken contract represents an LST to manage shares of a Vault in the form of a LSP7 token. There are restrictions for when a token is converted to and from LYX. Currently it is limited to only allow conversion when the contract is not paused (whenNotPaused) on onVaultStakeReceived and burn functions. If it is NOT expected to allow transfer of tokens while contract is paused, it is recommended to:

1. Remove burn function override
2. Remove whenNotPaused modifier from onVaultStakeReceived
3. Apply whenNotPaused modifier to _beforeTokenTransfer

Recommendation: Assumption of vault implementation

Revision: fixed

LiquidStakingToken contract makes assumptions about Vault implementation. In particular it assumes a vault is deployed as a proxy using openzeppelin TransparentUpgradeableProxy (ERC1967). If this assumption is broken and the vault is not ERC1967, it will prevent transfer of any tokens. It is recommended to:

1. Remove proxy implementation check in _beforeTokenTransfer, the affected line is: to == IProxy(address(stakingVault)).implementation()