AUDITISM

The art of audit

Lingo Audit Report

Version 1.0

Auditism

December 2, 2024

Lingo Audit Report

Auditism

November 29, 2024

Prepared by: Auditism

Table of Contents

Protocol Summary

LingoToken is an ERC20 token with a fee-on-transfer mechanism. The TokenStaking contract enables Lingo token holders to stake their tokens and register for off-chain rewards. The TokenVesting contract allows eligible users to claim their Lingo tokens according to a predefined vesting schedule.

Disclaimer

Auditism makes all effort to find as many vulnerabilities in the code in the given time period, but holds no responsibilities for the findings provided in this document. A security audit is not an endorsement of the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the Solidity implementation of the contracts.

Risk Classification

		Impact		
		High	Medium	Low
	High	H	H/M	M
Likelihood	Medium	H/M	M	M/L
	Low	M	M/L	L

Audit Details

Scope

LingoToken.sol
TokenStaking.sol
TokenVesting.sol

Executive Summary

Threat analysis

The logic of the Lingo protocol is straightforward. This audit primarily focused on ensuring that setter functions are properly secured with appropriate access controls. Additionally, the vesting claim logic was thoroughly analyzed to ensure that token amounts are calculated accurately, as this represents one of the major threat points requiring careful inspection, and to confirm that the MerkleProof implementation is correct. The staking mechanism was also rigorously reviewed to ensure it functions as intended.

Issues found

Medium risk 1
Low risk 3

Findings

Medium

M-01 User can bypass LingoToken transfer fees by staking with a 0 block period.

Description

When staking LingoTokens to the TokenStaking contract, the transfer fees will be exempted. Therefore it is possible to bypasse Lingo token transfer fees by staking with 0 block period and set the destination address as user in the arguments.

Mitigation Route

Consider restricting users to stake only for themselves. And only allow whitelisted roles to stake for someone else

```
function stake(

uint256 _amount,

uint256 _durationIndex,

uint256 _expectedDuration,

address _user

) external {
+ if(!LINGO_TOKEN.hasRole(LINGO_TOKEN.INTERNAL_ROLE(), msg.sender() && _user != msg.sender) revert notAllowed();
if (_amount < MIN_DEPOSIT) revert InsufficientAmount();
```

Low

L-01 Wrong error displayed

Description

Currently, when the staking contract doesn't have the internal role, the transaction will revert with `InsufficientAmount`. This error doesn't communicate the essence of the revert reason accurately and should be changed

Mitigation Route

Please add a new error describing more accurately the revert reason. An error name such as `noInteralRole` would be more descriptive.

L-02 Missing WhitelistUpdated event

Description

The docs mention the `WhiteListUpdated()` event however it hasn't been implemented in the function managing the whitelist. It is considered good practice to add events to setter() functions.

Mitigation Route

Add events to the `addExternalAccess()`, `addInternalAccess()` and `revokeAccess()` functions.

L-03 Lingo Token fees will not be applied for small token transfers

Description

When making transfers a fee will be applied to transfers if the contracts involved are not from the Lingo protocol. Given that the formula to calculate the fee is `amount * fee / 10_000` if the product of `amount * fee` < 10 000 no fees may be applied.

Example scenario:

`transferFee` is 50
`amount` is 199
`fee` would be $199 * 50/10_000 = 9950/10_000$ this amount is smaller than one and thus equates to 0 in Solidity.

Given the gas costs of any on chain transactions it may not be profitable to purposely create small transaction in order to avoid Lingo fees.

Mitigation Route

In order to avoid this bypass, consider adding one wei to any transaction.

```
uint256 fee = ((amount * transferFee)/ PERCENTAGE_DIVISOR)+ 1;
```