STATE MIND

TRP vesting escrow

24-01-2023 - 27-01-2023

Table of contents

1. Project Brief

Title	Description
Client	Lido
Project name	TRP vesting escrow
Timeline	24-01-2023 - 27-01-2023
Initial commit	dfe7bde89113825258190 48b3beda524b2c3a3bf
Final commit	69dd13adcd9c5a88da8c134b221209ccded04121

Short Overview

Token reward program (TRP) escrow contracts should allow transparent on-chain distribution and vesting of the token rewards for the Lido DAO contributors.

Project Scope

The audit covered the following files:

VestingEscrow.vy

VestingEscrowFactory.vy

VotingAdapter.vy

2. Finding Severity breakdown

All vulnerabilities discovered during the audit are classified based on its potential severity and has the following classification:

Severity	Description
Critical	Bugs leading to assets theft, fund access locking, or any other loss funds to be transferred to any party.
High	Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.
Medium	Bugs that can break the intended contract logic or expose it to DoS attacks, but do not cause direct loss funds.
Informational	Bugs that do not have a significant immediate impact and could be easily fixed.

Based on the feedback received from the Customer regarding the list of findings discovered by the Contractor, they are assigned the following statuses:

Status	Description
Fixed	Recommended fixes have been made to the project code and no longer affect its security.
Acknowledged	The Customer is aware of the finding. Recommendations for the finding are planned to be resolved in the future.

3. Summary of findings

Severity	# of Findings
Critical	0
High	0
Medium	0
Informational	5

4. Conclusion

Commit with all fixes: `69dd13adcd9c5a88da8c134b221209ccded04121`
No critical, high or medium severity issues were found.
5 informational severity issue were found, 4 out of 5 issues were fixed, 1 acknowledged.

Deployment

File name	Contract deployed on mainnet
VestingEscrow.sol	0x484FD04c598A09536ODF89bF85AB34c37127AA39
VestingEscrowFactory.sol	0xDA1DF6442aFD2EC36aBEa91029794B9b2156ADD0
VotingAdapter.sol	0xCFda8aB0AE5F4Fa33506F9C5165OB890E4871Cc1

5. Findings report

Informational

Redundant argument	Acknowledged

Description

In all contracts in the function `recover_erc20` argument `amount` can be removed. The function can transfer all balances like in the function `recover_ether`.

Recommendation

Argument `amount` can be removed in the function `recover_erc20`, but this will add one additional external call to the token of function `balanceOf`.

Client's comments

`amount` arg is crucial due to the existence of the "non-canonical" ERC20 tokens (https://github.com/d-xo/weird-erc20). For some implementations of the ERC20 `token.transfer(recipient, token.balanceOf(this))` might always fail. Nothing to fix or improve here.

Gas optimization	Fixed at c6de73

Description

There are some items in the VestingEscrow contract:

1. In the function `initialize`, there is a read of variable `self.token` after write
2. In the function `recover_erc20` global variable `self.token` is read twice (here and here).

Recommendation

Here and here replace `self.token` with `ERC20(token)`.

The function claim() doesn't return value claimed	Fixed at def810

Description

At the line contracts/VestingEscrow.vy#L179.

The function `claim()` transfers `claimable` amount of tokens to the beneficiary, but `claimable` can be smaller than `amount` parameter. If the recipient is a contract, it may be useful to return `claimable` value.

Recommendation

Consider returning `claimable` value in the function `claim()`.

No check if amount to be vested is bigger than 0	Fixed at 69dd13

Description

In the function `deploy_vesting_contract()` at the line contracts/VestingEscrowFactory.vy#L89.
The function parameter `amount` is not checked to be bigger than `0`. Therefore, it is possible to create escrows with no tokens.

Recommendation

It is recommended to check if `amount > 0`.

Inline block.timestamp	Fixed at aa55fb

Description

Internal function parameter `time` is never passed(_unclaimed and _locked).

Recommendation

Default value `block.timestamp` can be used instead of variable to optimize gas.

8. Appendix C. Tests

Tests result

```
131 passed, 26 warnings in 88.09s (0:01:28)
```

Tests coverage

Contracts

Contract	Coverage
VestingEscrow	88.9%
VestingEscrowFactory	100.0%
VotingAdapter	0%

Functions

Function	Coverage
VestingEscrow._check_sender_is_owner	100.0%
VestingEscrow._check_sender_is_owner_or_manager	100.0%
VestingEscrow._check_sender_is_recipient	100.0%
VestingEscrow._check_voting_adapter_is_set	100.0%
VestingEscrow._locked	100.0%
VestingEscrow._total_vested_at	100.0%
VestingEscrow._unclaimed	100.0%
VestingEscrow.revoke_all	93.8%
VestingEscrow.recover_erc20	91.7%
VestingEscrow.initialize	87.5%
VestingEscrow.revoke_unvested	87.5%
VestingEscrow.claim	75.0%
VestingEscrow.recover_ether	75.0%

Function	Coverage
VestingEscrow._safe_send_ether	37.5%
VestingEscrowFactory.deploy_vesting_contract	100.0%

STATE
MIND