



Organ Treasury Audit Report

Prepared by [T1MOH](#)

Version 1.0

Lead Auditors

[T1MOH](#)

September 29th, 2026

Contents

1 About T1MOH 2

2 Disclaimer 2

3 Risk Classification 2

4 Protocol Summary 2

5 Audit Scope 2

 5.0.1 organ-treasury 2

6 Executive Summary 2

7 Findings 4

 7.1 Low Risk 4

 7.1.1 A paused or blocklisted quote token bricks fee claiming, distribution and beneficiary migration 4

1 About T1MOH

T1MOH is an independent smart contract security researcher working with [Spearbit](#), [Cyfrin](#), [Sherlock](#), [BurraSec](#). Portfolio [can be found by link](#).

2 Disclaimer

T1MOH makes every effort to find as many vulnerabilities in the code as possible in the given time but holds no responsibility for the findings in this document. A security audit does not endorse the underlying business or product. The audit was time-boxed and the review of the code was solely focused on the security aspects of the Solidity implementation of the contracts.

3 Risk Classification

	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

4 Protocol Summary

The reviewed contracts include:

- `FeeCollector` - a non-custodial fee collector and splitter that acts as the beneficiary for Doppler Uniswap v4 pools. It claims LP and hook fees, burns the Organ token, and splits quote-asset fees on an immutable route (20% to Labs, 80% to Treasury). It also supports a time-delayed beneficiary migration.
- `Treasury` - accumulates protocol fees and rebalances them through owner-approved Uniswap v4 pools. Swaps are built on-chain (single-hop V4 swap via `UniversalRouter`, paid through `Permit2`), with output taken by the Treasury itself; any Organ bought back is burned in the same transaction and can never be sold.

5 Audit Scope

5.0.1 organ-treasury

Contract

`src/FeeCollector.sol`

`src/Treasury.sol`

6 Executive Summary

Over the course of 1 day, T1MOH conducted an audit on the [Organ Treasury](#) smart contracts provided by Organ. In this period, a total of 1 issues were found.

The findings consist of 1 Low severity issue.

Summary

Repository	organ-treasury
Commit	f60e53eed776...
Audit Timeline	Sep 29th, 2026
Methods	Manual Review

Issues Found

Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	1
Informational	0
Gas Optimizations	0
Total Issues	1

Summary of Findings

A paused or blocklisted quote token bricks fee claiming, distribution and beneficiary migration	
---	--

7 Findings

7.1 Low Risk

7.1.1 A paused or blocklisted quote token bricks fee claiming, distribution and beneficiary migration

Description: `FeeCollector._distribute()` pays out to both labs and treasury inside one atomic call, and every transfer is enforced to succeed:

```
// ERC20 path
IERC20(token).safeTransfer(labs, labsShare);
IERC20(token).safeTransfer(treasury, treasuryShare);

// native ETH path
(bool s1,) = labs.call{value: labsShare}("");
if (!s1) revert TransferFailed();
(bool s2,) = treasury.call{value: treasuryShare}("");
if (!s2) revert TransferFailed();
```

If either leg fails, the whole call reverts. Most trivial reason is that token is paused.

Recommended Mitigation: Decouple claiming from paying out, and make each payout leg independently failable. Consider switching the split to a pull pattern: accrue `owed[labs][token]` and `owed[treasury][token]` in `_distribute()` and expose a `withdraw(token, receiver)` that anybody can call.