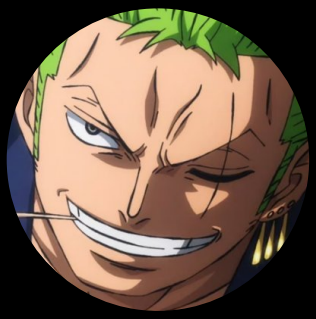
Tokensoft
Security Review

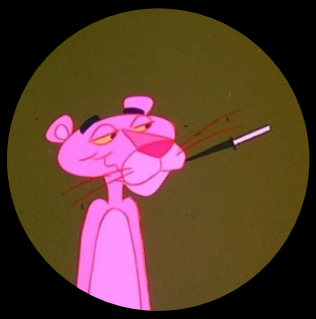
MAY, 2024

www.chaindefenders.xyz
https://x.com/ChDefendersEth

Lead Auditors



PeterSR



0x539.eth

Table of Contents

Protocol Summary

Adding “Per Address” functionality to existing distribution contracts on Tokensoft’s platform.

Disclaimer

The ChainDefenders team 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 by the team 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

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

Audit Details

Scope

Id	Files in scope
1	PerAddressContinuousVestingMerkle.sol
2	PerAddressTrancheVestingMerkle.sol
3	AdvancedDistributor.sol
4	Distributor.sol
5	MerkleSet.sol
6	PerAddressContinuousVesting.sol
7	PerAddressTrancheVesting.sol
8	AdvancedDistributorInitializable.sol
9	DistributorInitializable.sol
10	FairQueueInitializable.sol
11	MerkleSetInitializable.sol
12	PerAddressContinuousVestingInitializable.sol
13	PerAddressContinuousVestingMerkleDistributor.sol
14	PerAddressContinuousVestingMerkleDistributorFactory.sol
15	PerAddressTrancheVestingInitializable.sol
16	PerAddressTrancheVestingMerkleDistributor.sol
17	PerAddressTrancheVestingMerkleDistributorFactory.sol
18	IDistributor.sol
19	ITrancheVesting.sol
20	Registry.sol
21	Sweepable.sol

Roles

Id	Roles
1	User
2	Owner

Executive Summary

Issues found

Severity	Count	Description
High	0	Critical vulnerabilities
Medium	1	Significant risks
Low	0	Minor issues with low impact
Informational	0	Best practices or suggestions
Gas	0	Optimization opportunities

Findings

Medium

Mid 01 DoS In Claiming

Summary

The `claim` function in the `PerAddressContinuousVestingMerkleDistributor` contract uses `new bytes(0)` (an empty byte array) as hardcoded data. This causes the function to always revert when calling `getVestedFraction`, as it cannot process an empty byte array.

Vulnerability Detail

The `claim` function in the `PerAddressContinuousVestingMerkleDistributor` contract is hardcoded to use `new bytes(0)` (an empty byte array). This leads to a revert in the `getVestedFraction` function, which cannot handle an empty byte array.

Impact

This effectively breaks the `claim` functionality, making it a bug because users cannot claim their vested tokens.

Code Snippet

```
function claim(
    uint256 index, // the beneficiary's index in the merkle root
    address beneficiary, // the address that will receive tokens
    uint256 totalAmount, // the total claimable by this beneficiary
    uint256 start, // the start of the vesting period
    uint256 cliff, // cliff time
    uint256 end, // the end of the vesting period
    bytes32[] calldata merkleProof
)
external
validMerkleProof(keccak256(abi.encodePacked(index, beneficiary,
   totalAmount, start, cliff, end)), merkleProof)
nonReentrant
{
    // effects
    uint256 claimedAmount = super._executeClaim(beneficiary,
   totalAmount, new bytes(0));
    // interactions
    _settleClaim(beneficiary, claimedAmount);
}
```

Tool used

Manual Review

Recommendation

Modify the `claim` function to handle a valid byte array that can be processed by the `getVestedFraction` function.