THORWallet
Security Review

FEBRUARY, 2025

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

Lead Auditors

PeterSR

0x539.eth

Table of Contents

Protocol Summary

Thorwallet is an all-in-one DeFi solution that seamlessly combines cross-chain trading, fiat accessibility, and a powerful multichain multisignature solution—all in a single, intuitive platform.

Disclaimer

The Chain Defenders 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	MergeTgt.sol
2	Titn.sol
3	IERC677Receiver.sol
4	IMerge.sol

Roles

Id	Roles
1	Owner
2	User

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 Bridge Restriction Griefing

Finding description and impact

In `Titn` addresses that have bridged tokens will forever have transfer restrictions (until they are globally lifted). This means that once an address receives bridged tokens, it will be marked in the `isBridgedTokenHolder` mapping and will be subject to transfer restrictions indefinitely. A malicious actor can exploit this by sending a small amount of bridged tokens to any target address, thereby marking it as a bridged token holder and effectively blocking its ability to transfer tokens freely.

This can disrupt the normal functionality of the target address, especially if it is a contract or an address that requires unrestricted token transfers. What is more, such restriction is unfair and will lead to trust loss in the long run.

```
function _credit(
    address _to,
    uint256 _amountLD,
    uint32 /*_srcEid*/
) internal virtual override returns (uint256 amountReceivedLD) {
    if (_to == address(0x0)) _to = address(0xdead); // _mint(...) does not support address(0x0)
    // Default OFT mints on dst.
    _mint(_to, _amountLD);

    // Addresses that bridged tokens have some transfer restrictions
    if (!isBridgedTokenHolder[_to]) {
        isBridgedTokenHolder[_to] = true;
    }

    // In the case of NON-default OFT, the _amountLD MIGHT not be == amountReceivedLD.
    return _amountLD;
}
```

Proof of Concept

1. Malicious actor bridges from Arbitrum to Base a small amount of tokens to a target address (e.g., 1 wei of the token).
2. The target address is now marked in the `isBridgedTokenHolder` mapping.
3. The target address is subject to transfer restrictions, preventing it from transferring tokens freely.

Recommended Mitigation Steps

Implement a mechanism to remove addresses from the `isBridgedTokenHolder` mapping after a certain condition is met, such as a time period or an explicit action by the owner. This will ensure that such griefing attack is less likely to affect users negatively.