OpenZeppelin | security

Everdawn USDT0 ERC-7802 Upgrade Audit

March 21, 2025

Table of Contents

Summary

Type	DeFi/Stablecoin
Timeline	From 2025-03-10 To 2025-03-10
Languages	Solidity JavaScript TypeScript

Total Issues	3 (0 resolved)
Critical Severity Issues	0 (0 resolved)
High Severity Issues	0 (0 resolved)
Medium Severity Issues	0 (0 resolved)
Low Severity Issues	1 (0 resolved)
Notes & Additional Information	2 (0 resolved)

Scope

We diff-audited the Everdawn-Labs/usdt0-oft-contracts repository at pull request #63 and the Everdawn-Labs/usdt0-tether-contracts-hardhat repository at pull request #42.

The audit scope included the following files:

```
usdt0-tether-contracts-hardhat
└── contracts
    ├── Tether
    |   └── TetherToken.sol
    └── Wrappers
        ├── ArbitrumExtension.sol
        ├── HyperliquidExtension.sol
        ├── OFTExtension.sol
        └── interfaces
            └── IERC7802.sol
usdt0-oft-contracts
└── contracts
    ├── OUpgradeable.sol
    └── interfaces
        └──IERC7802.sol
```

We also reviewed the deployment scripts for the Corn network at pull request #68 of the Everdawn-Labs/usdt0-oft-contracts repository and pull request #48 of the Everdawn-Labs/usdt0-tether-contracts-hardhat repository.

The review scope included the following files:

```
usdt0-oft-contracts
├── hardhat.config.ts
├── layerzero-prod.config.ts
└── deploy
    └── ProductionDeployments.ts
usdt0-tether-contracts-hardhat
└── hardhat.config.js
```

System Overview

USDT0 is an ERC-20 wrapper for Tether (USDT) that integrates with LayerZero's Omnichain token format, called Omnichain Fungible Token (OFT). This allows simple and seamless cross-chain transfers from the token itself. The goal of these pull requests is to update USDT0 and its chain-specific implementations to conform to the draft ERC-7802 Crosschain Token Interface Standard.

Security Model and Trust Assumptions

This being a diff audit, we assume that the core interactions and assumptions of the contracts that were unchanged are effective, robust, and secure. This includes the LayerZero infrastructure and the greater portion of the USDT0 contract code. There is an `owner` role in the system that allows the changing of critical functionality in the contracts. We assume that this owner will act competently and in good faith.

Deployment Scripts for the Corn Chain Support

We reviewed the deployment scripts for the Corn chain and discovered the following.

Regarding the scripts in Everdawn-Labs/usdt0-oft-contracts repository:

- The Corn chain configuration as a LayerZero endpoint has been added properly (`hardhat.config.ts`).
  - Endpoint is added using the LayerZero's `lz-definitions` module.
  - *Safe API for the corn chain has been specified. However, the Safe account address is missing at the moment even though it has been created.*

- The Corn deployment configuration for the `OUpgradeable` contract has been added properly (`deploy/Productiondeployments.ts`):
  - The USDT0 token address in the Corn mainnet chain, which will be an initialization argument for `OUpgradeable`, is correct (`deploy/Productiondeployments.ts`).
- The Corn `OUpgradeable` contract has been added as an extra supported endpoint to the LayerZero configuration script. Connections with the rest of the currently supported endpoints have been set as well (`layerzero-prod.config.ts`).

Regarding the scripts in Everdawn-Labs/usdt0-tether-contracts-hardhat repository:

- The Corn chain configuration has been added properly (`hardhat.config.js`):
  - `chainId` is set correctly.

On-chain Deployment Review

Our team performed a review of the deployment scripts for both Unichain and Optimism mainnet deployments, as well as the upgrade of the `TetherTokenOFTExtension` and `OUpgradeable` contracts which is relevant to ERC-7802. These changes are included in pull request #51 at commit 75e33fd and pull request #73 at commit fa02ae3.

A bytecode level comparison between the compiled versions of the audited code and the deployed contracts found no discrepencies, further the Ink upgrade was reviewed and confirmed to have been properly configured and no storage collisions had been introduced.

In addition, we reviewed the scheduled transactions in the Safe wallets of each supported chain regarding the Unichain and Optimism endpoints support as well as the Ink upgrade.

For reference, the Safe wallet addresses for each one of the supported chains are listed below.

Chain	Safe Wallet Address	TXs Reviewed
Ethereum	0x4DFF9b5b0143E642a3F63a5bcf2d1C328e600bf8	31-39
Arbitrum	0x4DFF9b5b0143E642a3F63a5bcf2d1C328e600bf8	34-42
Ink	0xc95de55ce5e93f788A1Faab2A9c9503F51a5dAE2	29-38

Chain	Safe Wallet Address	TXs Reviewed
Flare	0x6ae078461f35c3cC216A71029F71ee7Bc4d9a10b	23-31
Berachain	0x425d1D17C33bdc0615eA18D1b18CCA7e14bEeb58	35-43
Corn	0x57d798f9d3B014bAC81A6B9fb3c18c0242A9411E	2-10
Unichain	0x4DFF9b5b0143E642a3F63a5bcf2d1C328e600bf8	5-28
Optimism	0x4DFF9b5b0143E642a3F63a5bcf2d1C328e600bf8	2-25

Low Severity

L-01 EIP-165 Conformity

EIP-165 specifies that the `supportsInterface` function must return `true` for all interfaces a contract implements. The `TetherTokenOFTExtension` and `ArbitrumExtensionFlattened` implementations will both return `true` for ERC-7802 and ERC-165, but they also implement ERC-20, ERC-3009, ERC-173, and ERC-2612 (`TetherTokenOFTExtension` only) for which they do not return `true`.

Consider returning `true` for the additional implemented interfaces as well.

Update: *Acknowledged, not resolved.*

Notes & Additional Information

N-01 Code Cleanliness

It is considered a good practice to have file names match the contract names and having one contract per file. This clarifies intention and makes the codebase easier to search/understand.

Consider renaming `OFTExtension.sol` to `TetherTokenOFTExtension.sol` and splitting `ArbitrumExtension.sol` into its constituent, self-named parts.

Update: *Acknowledged, not resolved. The team stated:*

> *Acknowledged, we would prefer not to split `ArbitrumExtension` to eliminate confusion with similarly named contracts.*

N-02 Missing Documentation

In `HyperliquidExtension.sol`, the `transferWithHop` function enables transferring ERC-20 tokens to HyperCore through HyperEVM. In addition, the owner can call the `setTrusted` function to whitelist any account which will be allowed to perform such transfers.

Consider adding documentation for `transferWithHop` as its functionality pertains to specific concepts of the Hyperliquid chain. In addition, consider documenting the entities which are expected to be registered as trusted via `setTrusted`.

Update: *Acknowledged, not resolved.*

Conclusion

We audited the recent changes made to Everdawn's USDT0 token that aim to make it compliant with ERC-7802. We also reviewed the deployment scripts for the Corn Chain support. One Low issue, regarding full compliance with ERC-165 standard, and a couple Note issues have been reported which can help improve the clarity and readability of the codebase. We are grateful to Everdawn as a partner and look forward to their project's success.