Smart Contract Security Audit Report

The SlowMist Security Team received the Trusta AI team's application for smart contract security audit of the TrustaOFT TA token on 2025.07.02. The following are the details and results of this smart contract security audit:

Token Name :

TrustaOFT TA token

The contract address :

https://github.com/TrustaLabs/TA_OFT

Initial audit commit: 8e71c13894f604a71299d015df169fa018825250

Review commit: 0b2d015f634dcdc92ad3a3ab37679030fc8e6603

The audit items and results :

(Other unknown security vulnerabilities are not included in the audit responsibility scope)

NO.	Audit Items	Result
1	Replay Vulnerability	Passed
2	Denial of Service Vulnerability	Passed
3	Race Conditions Vulnerability	Passed
4	Authority Control Vulnerability Audit	Some Risks
5	Integer Overflow and Underflow Vulnerability	Passed
6	Gas Optimization Audit	Passed
7	Design Logic Audit	Passed
8	Uninitialized Storage Pointers Vulnerability	Passed
9	Arithmetic Accuracy Deviation Vulnerability	Passed
10	"False top-up" Vulnerability	Passed
11	Malicious Event Log Audit	Passed
12	Scoping and Declarations Audit	Passed
13	Safety Design Audit	Passed

NO.	Audit Items	Result
14	Non-privacy/Non-dark Coin Audit	Passed

Audit Result : Medium Risk

Audit Number : 0X002507040003

Audit Date : 2025.07.02 - 2025.07.04

Audit Team : SlowMist Security Team

Summary conclusion : This is the Token contract for the TrustaOFT protocol. The total supply of TA tokens is fixed on main chain, the following excessive privilege risks should be noted:

The `_delegate` can setEnforcedOptions/setPeer/setDelegate/setPreCrime/setMsgInspector.

The source code:

- contracts/TrustaOFT.sol

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.22;

import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol";
import { Ownable } from "@openzeppelin/contracts/access/Ownable.sol";
import { OFT } from "@layerzerolabs/oft-evm/contracts/OFT.sol";


contract TrustaOFT is OFT, ERC20Permit {
    constructor(
        uint256 _mainChainId,
        string memory _name,
        string memory _symbol,
        address _lzEndpoint,
        address _delegate,
        address _treasury
    ) OFT(_name, _symbol, _lzEndpoint, _delegate) Ownable(_delegate)
ERC20Permit(_name) {
        if (block.chainid == _mainChainId) {
            _mint(_treasury, 1_000_000_000 * 10 ** 18);
        }
    }
}
```

Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.

Official Website

www.slowmist.com

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist