Cyberscope

A TAC Security Company

Audit Report

Plutus

June 2025

Repository

https://github.com/PlutusDao/plutus_multichain/tree/main

Commit e42b33144f82c8b72bf9a19ea6d810384127fd23

Audited by © cyberscope

Analysis

Critical Medium Minor / Informative Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Acknowledged
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	OCC	OFT Centralization Configuration	Acknowledged
●	MC	Missing Check	Acknowledged
●	L04	Conformance to Solidity Naming Conventions	Acknowledged

Table of Contents

Risk Classification

The criticality of findings in Cyberscope's smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
Critical	Highly Likely / High Impact
Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
Minor / Informative	Unlikely / Low to no Impact

Review

Repository	https://github.com/PlutusDao/plutus_multichain/tree/main
Commit	e42b33144f82c8b72bf9a19ea6d810384127fd23

Audit Updates

Initial Audit	02 Jun 2025

Source Files

Filename	SHA256
PlutusTokenOFT.sol	f56305d065541eacb1e4cd52d7eaf7de1ecd9e07e43c18cb388d50183ecc99c3
PlutusToken.sol	407ca3bc79fc674d60e66052ab61c9c74cc17ec52e4d0c82da93ec240af4aa5d

Overview

`PlutusToken` and `PlutusTokenOFT`, both represent tokens that inherit from the `OFT` (Omnichain Fungible Token) contract and in case of `PlutusToken` it also inherits from OpenZeppelin's `Ownable` contract for secure ownership management. `PlutusToken` is designed with a minting function that allows the contract owner to mint new tokens, while enforcing a maximum supply cap of `130` million tokens to prevent exceeding the predefined limit. In contrast, `PlutusTokenOFT` is meant for deployment on satellite chains and does not include the minting function, focusing solely on the basic token functionality with the same structure as the main token. Both tokens leverage LayerZero's cross-chain capabilities, with `PlutusToken` operating on the primary chain and `PlutusTokenOFT` tailored for satellite chains.

Findings Breakdown

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	1	0	0
Medium	0	1	0	0
Minor / Informative	0	2	0	0

MT - Mints Tokens

Criticality	Medium
Location	PlutusToken.sol#L18
Status	Acknowledged

Description

The contract owner has the authority to mint tokens. The owner may take advantage of it by calling the `mint` function. As a result, the contract tokens will be highly inflated.

```
uint256 public constant MAX_SUPPLY = 130_000_000e18;
function mint(address _to, uint256 _amount) public onlyOwner {
    if (totalSupply() + _amount > MAX_SUPPLY) revert
MAX_SUPPLY_EXCEEDED();

    _mint(_to, _amount);
}
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

Team Update

The team has acknowledged that this is not a security issue and states: *The owner will be a multi-sig address. We have a DAO controlled core SafeWallet requiring 4 out of 7 cold wallet signatures to act. The cold wallet sigs must belong to 7 core Plutus team members chosen by the DAO both as signers and as employees. So the protocol is both de-centralised and trusted.*

OCC - OFT Centralization Configuration

Criticality	Critical
Location	PlutusToken.sol#L5 PlutusTokenOFT.sol#L5
Status	Acknowledged

Description

The owner of both `PlutusToken` and `PlutusTokenOFT` can change their cross-chain configuration with functions like `setPeer`. These changes are also possible post configuration. As a result this creates a centralization risk where users will have to depend on the actions of the owner. If the owner alters these configurations they can disrupt users' cross-chain operations.

```
import { OFT } from "@layerzerolabs/oft-evm/contracts/OFT.sol";
```

Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the ownership, which will eliminate the threats but it is non-reversible.

Team Update

The team has acknowledged that this is not a security issue and states: *The owner is going to be a multi-sig to manage these administrative actions. We have a DAO controlled core SafeWallet requiring 4 out of 7 cold wallet signatures to act. The cold wallet sigs must belong to 7 core Plutus team members chosen by the DAO both as signers and as employees. So the protocol is both de-centralised and trusted.*

MC - Missing Check

Criticality	Minor / Informative
Location	PlutusTokenOFT.sol#L10,11 PlutusToken.sol#L12,13
Status	Acknowledged

Description

The contract is processing variables that have not been properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

Specifically, in both contracts' constructors it is not checked if the strings provided for `name` and `symbol` are empty.

```
constructor(
    string memory _name,
    string memory _symbol,
    address _lzEndpoint,
    address _delegate
) ...
```

Recommendation

The team is advised to properly check the variables according to the required specifications.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	PlutusToken.sol#L18
Status	Acknowledged

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
uint256 _amount
address _to
```

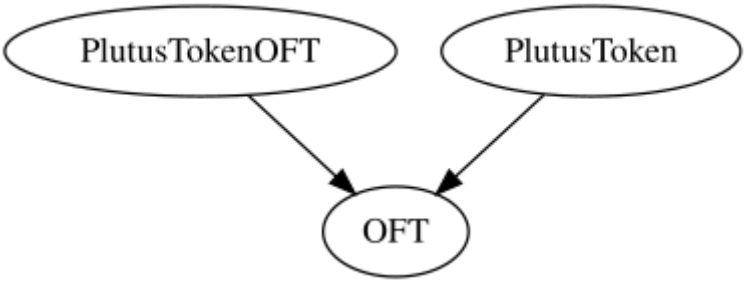
Recommendation

By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/stable/style-guide.html#naming-conventions.

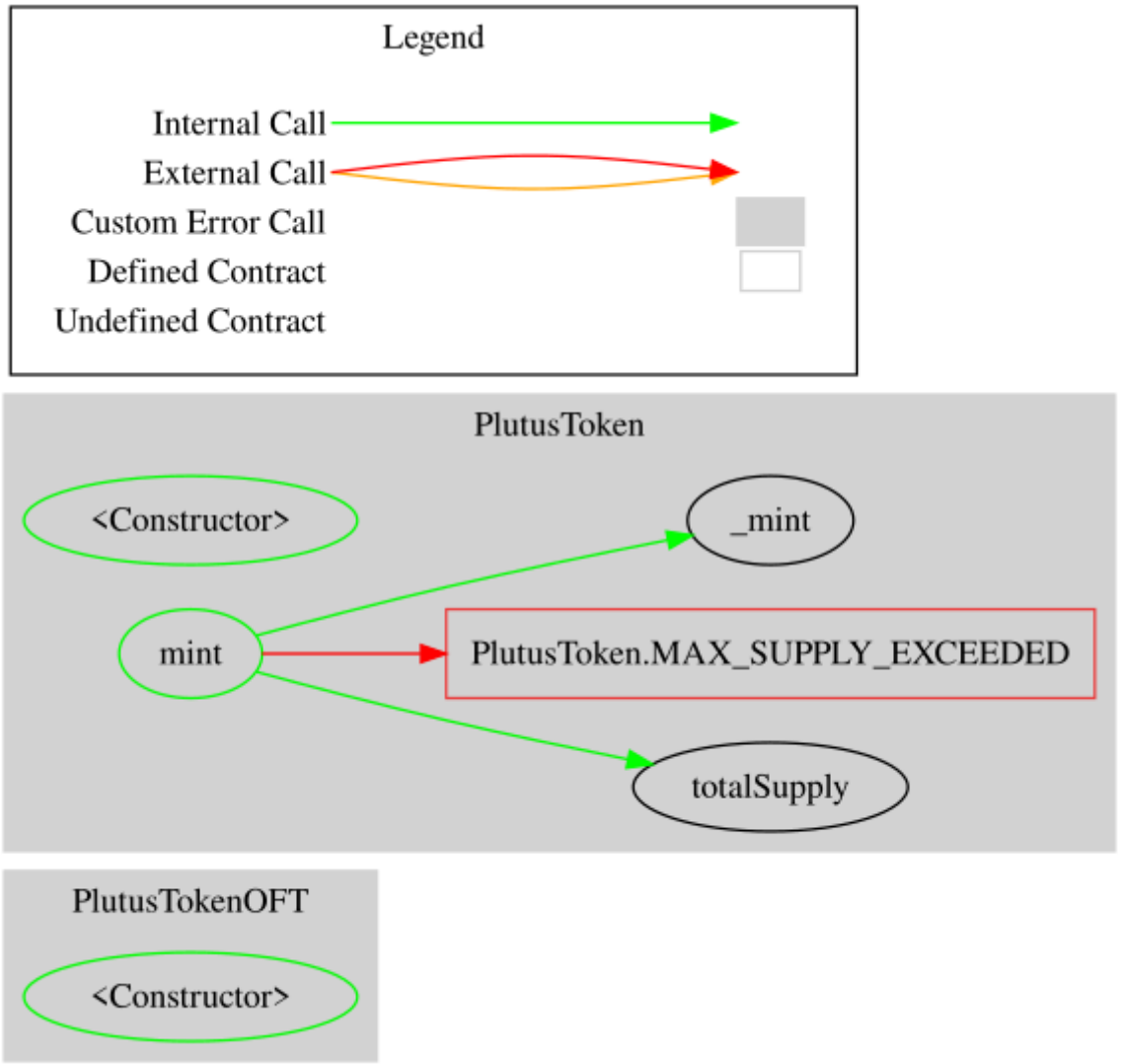
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
PlutusTokenOFT	Implementation	OFT		
		Public	✓	OFT Ownable
PlutusToken	Implementation	OFT		
		Public	✓	OFT Ownable
	mint	Public	✓	onlyOwner

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
PlutusToken
<Constructor>
_mint
mint
PlutusToken.MAX_SUPPLY_EXCEEDED
totalSupply
PlutusTokenOFT
<Constructor>

Summary

Plutus contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. There are some functions that can be abused by the owner like mint tokens. If the contract owner abuses the mint functionality, then the contract will be highly inflated. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats. The team has acknowledged all findings.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

A **TAC Security** Company

The Cyberscope team

cyberscope.io