PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Preliminary Report

For Usual (USD0)

11 October 2024

paladinsec.co

info@paladinsec.co

Table of Contents

Table of Contents 2

Disclaimer 3

1 Overview 4

1.1 Summary 4

1.2 Contracts Assessed 4

1.3 Findings Summary 5

1.3.1 L2Usd0 6

1.3.2 L2Usd0PP 6

1.3.3 L1OFTAdapter 6

1.3.4 OFTMintAndBurnAdapter 6

2 Findings 7

2.1 L2Usd0 7

2.1.1 Privileged Functions 7

2.1.2 Issues & Recommendations 8

2.2 L2Usd0PP 11

2.2.1 Privileged Functions 11

2.2.2 Issues & Recommendations 12

2.3 L1OFTAdapter 14

2.3.1 Privileged Functions 14

2.3.2 Issues & Recommendations 14

2.4 OFTMintAndBurnAdapter 15

2.4.1 Privileged Functions 15

2.4.2 Issues & Recommendations 15

Disclaimer

Paladin Blockchain Security ("Paladin") has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Paladin.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and Paladin is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Paladin or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

Paladin retains the right to re-use any and all knowledge and expertise gained during the audit process, including, but not limited to, vulnerabilities, bugs, or new attack vectors. Paladin is therefore allowed and expected to use this knowledge in subsequent audits and to inform any third party, who may or may not be our past or current clients, whose projects have similar vulnerabilities. Paladin is furthermore allowed to claim bug bounties from third-parties while doing so.

1 Overview

This report has been prepared for Usual's USD0 contracts on the Arbitrum network. Paladin provides a user-centred examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

1.1 Summary

Project Name	Usual
URL	https://usual.money
Platform	Arbitrum
Language	Solidity
Preliminary Contracts	https://github.com/usual-dao/pegasus/tree/d719bd984bb53745d534a6143514c136a3031dc1/packages/solidity https://github.com/LayerZero-Labs/usual-usd0-oft/tree/a74ff9adcedf22fce006826acabac39f854a4f28/contracts
Resolution	https://github.com/usual-dao/pegasus/tree/cb73fb1b643f804b173d50b6571fd75a602ea980

1.2 Contracts Assessed

Name	Contract	Live Code Match
`L2Usd0`		
`L2Usd0PP`		

1.3 Findings Summary

Severity	Found	Resolved	Partially Resolved	Acknowledged (no change made)
Governance	2	-	-	2
High	0	-	-	-
Medium	0	-	-	-
Low	1	1	-	-
Informational	5	5	-	-
Total	**8**	**6**	-	**2**

Classification of Issues

Severity	Description
Governance	Issues under this category are where the governance or owners of the protocol have certain privileges that users need to be aware of, some of which can result in the loss of user funds if the governance's private keys are lost or if they turn malicious, for example.
High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
Medium	Bugs or issues that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
Informational	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

1.3.1 L2Usd0

ID	Severity	Summary	Status
01	GOV	Governance privileges	ACKNOWLEDGED
02	INFO	Typographical issues	✓ RESOLVED
03	INFO	Missing amount 0 check in burn functions	✓ RESOLVED
04	INFO	Special role for blacklisting can be added	✓ RESOLVED

1.3.2 L2Usd0PP

ID	Severity	Summary	Status
05	GOV	Governance Privileges	ACKNOWLEDGED
06	LOW	The Pausable dependency is not initialized	✓ RESOLVED
07	INFO	Typographical issues	✓ RESOLVED
08	INFO	Missing amount 0 check in burn functions	✓ RESOLVED

1.3.3 L1OFTAdapter

No issues found.

1.3.4 OFTMintAndBurnAdapter

No issues found.

2 Findings

2.1 L2Usd0

`L2Usd0` is a basic token that can be minted and burned by accounts that have the `USD0_MINT` or `USD0_BURN` role.

This contract can also be paused to prevent any transfer, mint and burn by accounts that have the `PAUSING_CONTRACT_ROLE` .

Admins with the `DEFAULT_ADMIN_ROLE` can also blacklist specific addresses, preventing them from transferring the token.

This token will be minted and burned by the LayerZero OFT Adapter which will allow users to bridge their `USD0` from the main L1 to L2s using LayerZero.

2.1.1 Privileged Functions

- `pause [PAUSING_CONTRACTS_ROLE]`
- `unpause [PAUSING_CONTRACTS_ROLE]`
- `mint [USD0_MINT]`
- `burnFrom [USD0_BURN]`
- `burn [USD0_BURN]`
- `blacklist [DEFAULT_ADMIN_ROLE]`
- `unblacklist [DEFAULT_ADMIN_ROLE]`

2.1.2 Issues & Recommendations

Issue #01	Governance privileges
Severity	GOVERNANCE
Description	The contract allows anyone that have the right role to mint and burn tokens to and from users directly. This role is aimed to be given to the OFT Adapter only, but it can technically be given to other addresses. In the event of the admin key getting leaked, this could allow a malicious user to mint, burn and blacklist anyone.
Recommendation	Consider using a Gnosis multi-signature wallet as the default admin of this contract to reduce the impact of a key getting stolen.
Resolution	ACKNOWLEDGED A multi-signature wallet will be the default admin.

Issue #02	Typographical issues
Severity	INFORMATIONAL

Description	Lines 128 && 136
	`/// @dev Can only be called by the admin.`
	`/// @dev Can only be called by the admin.`
	
	The comment is misleading as it can only be called by users that have the `PAUSING_CONTRACTS_ROLE`.
	
	—
	
	Line 183
	`//@notice @Chainlink, .isContract from OZ4 was deprecated, which is why we replaced it with code.length;`
	
	The comment seems a bit ambiguous as Chainlink is mentioned.
	
	—
	
	`ITokenMapping tokenMapping;` from `Usd0StorageV0` seems to be unused.
	
	—
	
	Missing NatSpec param on `_createUsd0Checkfor registryContract`.

Recommendation	Consider fixing the typographical issues.
Resolution	RESOLVED

Issue #03	Missing amount 0 check in burn functions
Severity	INFORMATIONAL
Description	The mint function checks if the amount that must be minted is greater than 0, but the same check is missing from the burn functions.
Recommendation	Consider adding the same check to the burn functions.
Resolution	RESOLVED

Issue #04	Special role for blacklisting can be added
Severity	INFORMATIONAL
Description	Throughout the contract, a special role is set for every action but for the blacklist/unblacklist operations, the default admin is authorized. While this is generally fine, it is usually recommended to let the default admin to deal only with grant/revoke roles and not have extra assigned permissions to it.
Recommendation	Consider adding an extra role called `USD0_BLACKLIST`.
Resolution	RESOLVED A BLACKLIST_ROLE was added.

2.2 L2Usd0PP

L2USD0PP is a basic token that can be minted and burned by accounts that have the USD0PP_MINT or USD0PP_BURN role.

This contract can also be paused to prevent any transfer, mint and burn by accounts that have the PAUSING_CONTRACT_ROLE .

Users who are blacklisted in the L2USD0 token will also be blacklisted in USD0PP.

This token will be minted and burned by the LayerZero OFT Adapter which will allow users to bridge their USD0PP from the main L1 to L2s using LayerZero.

2.2.1 Privileged Functions

- pause [PAUSING_CONTRACTS_ROLE]
- unpause [DEFAULT_ADMIN]
- mint [USD0PP_MINT]
- burnFrom [USD0PP_BURN]
- burn [USD0PP_BURN]

2.2.2 Issues & Recommendations

Issue #05	Governance privileges
Severity	GOVERNANCE
Description	The contract allows anyone with the right role to mint and burn tokens to and from users directly. This role is aimed to be given to the OFT Adapter only, but it can technically be given to other addresses. In the event of the admin key getting leaked, this could allow a malicious user to mint, burn and blacklist anyone.
Recommendation	Consider using a Gnosis multi-signature wallet as the default admin of this contract to reduce the impact of a key getting stolen.
Resolution	ACKNOWLEDGED A multi-signature wallet will be the default admin.

Issue #06	The Pausable dependency is not initialized
Severity	LOW SEVERITY
Description	The contract does not initialize the Pausable dependency, however this issue is only rated low because the Pausable dependency does not create any issues if it is not initialized. However, it is still considered best practice to always initialize all the dependencies.
Recommendation	Consider intiliazing the Pausable dependency.
Resolution	RESOLVED

Issue #07	Typographical issues
Severity	INFORMATIONAL
Description	Line 176 `//@notice @Chainlink, .isContract from OZ4 was deprecated, which is why we replaced it with code.length;` The comment seems a bit ambiguous as Chainlink is mentioned. — Missing NatSpec param on `_createUsd0Checkfor registryContract`. — `IERC677Receiver` is imported twice, and the `IERC20` import is obsolete. The variable `usd0` can be `IUsd0`.
Recommendation	Consider fixing the typographical issues.
Resolution	RESOLVED

Issue #08	Missing amount 0 check in burn functions
Severity	INFORMATIONAL
Description	The mint function checks if the amount that must be minted is greater than 0, the same check is missing from the burn functions.
Recommendation	Consider adding the same check to the burn functions.
Resolution	RESOLVED

2.3 L1OFTAdapter

`L1OFTAdapter` allows users to lock their tokens in the adapter to then bridge them to a L2 using LayerZero.

This contract will be used with USD0 and USD0PP to allow users to bridge their USD0 and USD0PP tokens from L1 to the different L2s using Layer Zero.

2.3.1 Privileged Functions

- `setRateLimits`
- `setMsgInspector`
- `setPeer`
- `setDelegate`
- `setPreCrime`

2.3.2 Issues & Recommendations

No issues found with respect to the interactions with the USD0 and USD0PP contracts.

2.4 OFTMintAndBurnAdapter

`OFTMintAndBurnAdapter` is an adapter that facilitates the plugging of a mint and burn token on top of the LayerZero OFT standard.

This contract will be used with L2USD0 and L2USD0PP to allow users to bridge their USD0 and USD0PP from the L1 to the different L2s using Layer Zero.

2.4.1 Privileged Functions

- `setRateLimits`
- `setMsgInspector`
- `setPeer`
- `setDelegate`
- `setPreCrime`

2.4.2 Issues & Recommendations

No issues found with respect to the interactions with the USD0 and USD0PP contracts.


PALADIN
BLOCKCHAIN SECURITY