PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Final Report

For WINR Protocol (Whitelist Pool)

03 March 2023

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 WhitelistPool 6

2 Findings 7

2.1 WhitelistPool 7

2.1.1 Privileged Functions 7

2.1.2 Issues & Recommendations 8

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 WINR Protocol's Whitelist Pool contract 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	WINR Protocol
URL	https://winr.games/
Platform	Arbitrum
Language	Solidity
Preliminary Contracts	https://github.com/WINRLabs/winr-protocol/blob/b7f469c79a8a4787e3fcc4d3aad5716aa6e2c37b/contracts/tokens/vesting/WhitelistPool.sol
Final Contracts	https://github.com/WINRLabs/winr-protocol/blob/ff630f790fa0cbf1010bbfcc6d5ec91a77d6dd5b/contracts/tokens/vesting/WhitelistPool.sol

1.2 Contracts Assessed

Name	Contract	Live Code Match
`WhitelistPool`		

1.3 Findings Summary

Severity	Found	Resolved	Partially Resolved	Acknowledged (no change made)
High	0	-	-	-
Medium	1	1	-	-
Low	2	1	-	1
Informational	2	2	-	-
Total	**5**	**4**	**-**	**1**

Classification of Issues

Severity	Description
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 with 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 WhitelistPool

ID	Severity	Summary	Status
01	MEDIUM	Users can still deposit even after the `dueDate`	RESOLVED
02	LOW	Users will appear to have very little gWLP tokens due to a decimal error	RESOLVED
03	LOW	Governance risk: Contract admins can withdraw all USDC from the `WhitelistPool`	ACKNOWLEDGED
04	INFO	Typographical error	RESOLVED
05	INFO	Gas optimization	RESOLVED

2 Findings

2.1 WhitelistPool

`WhitelistPool` represents a component of the Winr presale where presale participants can exchange their USDC tokens for gWLP (Genesis WLP tokens).

The `deposit` function allows users to exchange any amount of USDC for an equivalent amount of gWLP. Meanwhile, the `withdraw` function allows the contract administrators to take out this USDC after the due date has passed. The due date is configured a few days after the deployment.

The due date can be updated via the `updateDueDate` function, however, they cannot shorten it. This means that they can only extend the due date.

2.1.1 Privileged Functions

- `withdraw [ DEFAULT_ADMIN_ROLE ]`
- `updateDueDate [ DEFAULT_ADMIN_ROLE ]`

2.1.2 Issues & Recommendations

Issue #01	Users can still deposit even after the `dueDate`
Severity	MEDIUM SEVERITY
Description	The WhitelistPool contract has a due date set at a fixed number of days after the pool has been deployed. However, there is currently no logic that prevents users from exchanging USDC for `gWLP` after this date has been reached. This means that users can continue to purchase `gWLP` forever.
Recommendation	Consider whether this is desired. If not, consider adding a requirement. `require(block.timestamp < dueDate, "WP: Due Date has passed");`
Resolution	RESOLVED The recommended requirement has been implemented.

Issue #02	**Users will appear to have very little gWLP tokens due to a decimal error**
Severity	LOW SEVERITY
Description	The WhitelistPool token, `gWLP`, has 18 decimals. However, the USDC which users deposit has 6 decimals. This causes the conversion to exchange a single USDC for a very small apparent amount of `gWLP`.

Recommendation	Consider adding an immutable `_decimals` variable:
	
	`uint256 private immutable _decimals;`
	
	`constructor(IERC20Metadata _USDC) … {`
	`  _decimals = _USDC.decimals();`
	`}`
	
	`function decimals() external override view returns (uint8) {`
	`  return _decimals;`
	`}`

Resolution	RESOLVED The client has taken a different approach by minting a whole 1e18 token for every 1e6 USDC which is deposited. This works as well and is valid.

Issue #03	Governance risk: Contract admins can withdraw all USDC from the `WhitelistPool`
Severity	LOW SEVERITY
Description	The contract admins can call `withdraw` to withdraw all the collected USDC once the `dueDate` expires. As there is no way for users to request a refund before the `dueDate`, the USDC could be lost if the governance keys are compromised or governance turns malicious. Another governance risk for the team itself is accidentally calling `updateDueDate` with a very large number (e.g. milliseconds) as this would effectively lock the team out of all USDC in the contract.
Recommendation	Consider whether it makes sense to allow users to exchange their gWLP for USDC again before the `dueDate`. This might not make sense for the tokenomics, however, we strongly urge the team to exclusively grant the `DEFAULT_ADMIN_ROLE` to a reputable multi-signature wallet with a minimum quorum of 3 unique and reputable parties. We also recommend adding a requirement that `newDueDate` is not too far in the future to avoid accidental lockouts.
Resolution	ACKNOWLEDGED The team has indicated they will hand over ownership to a multi-signature wallet after deployment.

Issue #04	Typographical error
Severity	INFORMATIONAL
Location	Line 52 `* @notice new due date can not be before than the current one`
Description	The word *than* can be removed from the sentence.
Recommendation	Consider fixing the typographical error.
Resolution	RESOLVED

Issue #05	Gas optimization
Severity	INFORMATIONAL
Location	Line 14 `IERC20 public USDC;`
Description	This token can be marked as immutable to save on gas usage.
Recommendation	Consider implementing the gas optimization mentioned above.
Resolution	RESOLVED

PALADIN
BLOCKCHAIN SECURITY