PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Final Report

For PolyBeta Finance

29 September 2021

paladinsec.co

info@paladinsec.co

Table of Contents

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.

1 Overview

This report has been prepared for PolyBeta Finance on the Polygon 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	PolyBeta Finance
URL	https://polybeta.finance/
Platform	Polygon
Language	Solidity

1.2 Contracts Assessed

Name	Contract	Live Code Match
`BETAToken`	`0xaC3090B7042FCA2cDBF233022e4a9823a032600c`	✓ MATCH
`BetaMasterChef`	`0x9581EA83B4BCd5F2c5f1705382FBd80a11E57DcD`	✓ MATCH

1.3 Findings Summary

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

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 BetaToken

ID	Severity	Summary	Status
01	LOW	mint function can be used to pre-mint large amounts of tokens before ownership is transferred to the Masterchef	RESOLVED

1.3.2 PolyBetaMasterChef

ID	Severity	Summary	Status
02	LOW	Contract uses raw subtraction for arithmetic operations	RESOLVED

2 Findings

2.1 BetaToken

The BetaToken is a simple ERC20 token.

2.1.1 Token Overview

Address	`0xaC3090B7042FCA2cDBF233022e4a9823a032600c`
Token Supply	15,600
Decimal Places	18
Transfer Max Size	None
Transfer Min Size	None
Transfer Fees	None
Pre-mints	1,600

2.1.2 Privileged Roles

The following functions can be called by the owner of the contract:

- `mint`

2.1.3 Issues & Recommendations

Issue #01	**`mint` function can be used to pre-mint large amounts of tokens before ownership is transferred to the Masterchef**
Severity	LOW SEVERITY
Description	The `mint` function could be used to pre-mint tokens for legitimate uses including, but not limited to, the injection of initial liquidity, token presale, or airdrops; however, this function may also be used to pre-mint tokens for dumping.
Recommendation	Consider being forthright if this `mint` function has been used by letting your community know how much was minted, where they are currently stored, if a vesting contract was used for token unlocking, and finally the purpose of the mints.
Resolution	RESOLVED The team pre-minted 1,600 Beta tokens and ownership has been transferred to the PolyBetaMasterChef. 1,528 Beta tokens out of the 1,600 that were pre-minted have been burned.

2.2 PolyBetaMasterChef

The BetaMasterChef contract was forked from PolyAlpha, which was previously audited by Paladin. As such, it is a battle-tested and secure Masterchef. Most notably, there are no hard rug risk functionalities within the contract. Deposit fees have an upper limit of 4%, and the upper limit of 15,600 tokens is enforced via the try/catch implementation in the `updatePool` function.

2.2.1 Privileged Roles

The following functions can be called by the owner of the contract:

- `add`
- `set`
- `setDevAddress`
- `setFeeAddress`
- `updateEmissionRate`
- `updateStartBlock`

2.2.2 Issues & Recommendations

Issue #02	Contract uses raw subtraction for arithmetic operations
Severity	LOW SEVERITY
Description	There is a risk of overflows because the contract uses raw subtraction with Solidity version 0.6.12. This can be found on Line 1235: `_amount = pool.lpToken.balanceOf(address(this)) - balanceBefore;` Note that this issue is also present in PolyAlpha.
Recommendation	Consider using SafeMath's sub rather than raw subtraction. Alternatively, upgrading to Solidity version 0.8.0 or higher would also solve this as SafeMath is implemented.
Resolution	RESOLVED

PALADIN
BLOCKCHAIN SECURITY