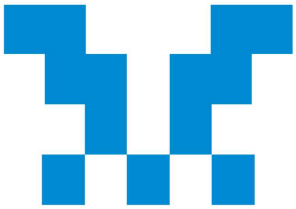
Ballena.io

DeFi Project Security Audit

Audit tool used:



MythX

Audit date: April - May 2021

Auditor: Javier Calatrava (linkedin.com/in/javiercalatrava)

VERSION CONTROL

Version	DATE	Main Changes	Author
0.1	12th April 2021	First audit, Initial Draft	JC
1	21st May 2021	Initial version, Migration phase	JC

CONTENT TABLE

Content table

ABSTRACT

This is the executive summary of the audit report based on performed analysis, in accordance with best cybersecurity and code development practices as at the date of this report, in relation to cybersecurity, vulnerabilities and issues in the framework and algorithms based on smart contracts and DeFi platform development.

This report must not be considered, under any circunstance, as an investment advice, as its only intention is to asure the security of the platform and the strongness of provided Smart Contracts.

Ballena.io started as an upgraded fork from beefy project (original v1 of the project), with specific Smart Contract developed within construction phase.

Ballena.io community has developed a DeFi platform for the community. The management is a DAO (Decentralized Autonomous Organization) structure, which is better to avoid fraudulent actions from the project owners.

On top of this, Ballena.io has applied multi-sign to the treasury and pools management smart contracts, avoiding the single point of failure or a weakness point of trustworthy, due to have a sole owner of the smart contracts.

Ballena.io is using the Gnosis Safe multi-signature system on the Binance Smart Chain with three (3) different wallets, depending on the requirements (https://docs.ballena.io/dao-organizacion/gnosis):

- Governance protocol. The most critical wallet out of the Ballena.io Gnosis system. This will be in charge of daily actions within the platform and will be allowed to operate with all the smart contracts. It will use a 6/9 model, so 6 out of 9 signatures are required to confirm an action.
- Operations protocol. This will be in charge of less critical operations but still requiring a minimum number of signatures to not compromise the project security level. Actions like change the rewarding multiplier, re-start a vault or agregate a new vault are part of the normal operations protocol. It will use a 6/9 model, so 6 out of 9 signatures are required to confirm an action.
- Security emergency protocol. This will be reserved for emergency actions, but not transcendental actions impacting the project. Pause a vault or the token rewards are the actions that this wallet can confirm, mainly to avoid a dangerous situation like an attack or any other issue that could be mitigated by pausing the vaults or rewards. As said, is only pausing, so can't delete

vaults, create new vaults or any other action that must be approved by several administrators. Only 1 out of 9 signatures is required for these actions.

- Last but not least, there is also a Donations protocol to manage the donations to be done from Ballena.io to external partners. This wallet can't interact with any contract within the project. It will follow as well a 6/9 model. 6 signatures out of 9.

By implementing the multi signature governance model, Ballena.io is providing a higher level of trustworthy from their administration community. It is a community project for the community users.

DISCLAIMER / LEGAL NOTICE

The data on this report is for your information only. It does not constitute investment advice, or advice on tax or legal matters.

Any information provided on this website does not constitute investment advice or investment recommendation nor does it constitute an offer to buy or sell or a solicitation of an offer to buy or sell shares or units in any of the investment funds or other financial instruments described on this website. In addition, the information provided on this website does not contain any offer, recommendation or incitement to conclude any contracts for financial services (e.g. wealth management) or to conclude any other kind of contract (e.g. Family Office agreements). In particular, this information should not be used as a substitute for suitable investment and product-related advice. Unless expressly stated otherwise, all pricing information is non-binding.

Should you have any doubts about the meaning of the information provided herein, please contact your financial advisor or any other independent professional advisor.

Background

Following smart contracts have been audited within the scope of this audit, any new contract or change in audited contracts will be documented in future versions of this document:

- ✔ *https://github.com/ballena-io/ballena-protocol/blob/master/contracts/token/BalleMigration.sol*

  Corresponding SC: https://bscscan.com/address/0x019e4abb7fd239f838d7ecbdd6601e0fcfcf497b#code

- ✔ *https://github.com/ballena-io/ballena-protocol/blob/master/contracts/token/BALLEv2.sol*

  Corresponding SC: https://bscscan.com/address/0x9714C04b34E6300964161c3aC37b86451E79152d#code

- ✔ *https://github.com/ballena-io/ballena-protocol/blob/master/contracts/vaults/BalleMaster.sol*

  Corresponding SC: (TestNET: https://testnet.bscscan.com/address/0x79c56bb227fe9c52ec7a8e78be86dd782ac5f023#code):

- ✔ *https://github.com/ballena-io/ballena-protocol/blob/master/contracts/strategies/StratPancakeLpV1.sol*

- ✔ Corresponding SC: (TestNET: https://testnet.bscscan.com/address/0xcbd94aac8d55e574150047b187312c982efb905c#code):

The application code have been audited as well.

- ✔ *https://github.com/ballena-io/ballena-app*
  Current status: CORRECT

Audit process has been done in several phases, initially to allow the Ballena.io development team to solve the potential code issues. Later, to chect the “almost final” version and finally, to ensure that code is robust and there are no issues impacting the users and their investments.

Within the initial phase of audit, few minor issues were found on javascript code, these issues were solved for the pre-production testing phase audit.

Some of these issues were inherit from the forked project.

Regarding the Smart Contracts, only few issues were found and all the recommendations have been applied.

Audit scope

- Audit of smart contracts to avoid negative impact on users.
- Audit of application code to avoid potential security vulnerabilities.

This audit is verifying the following audits points (with current status):

- Deny of Service attacks based on REVERT actions. CORRECT
- Deny of Service attacks based on gas limitation. CORRECT
- Warnings at compilation time. CORRECT
- Reentrancy and race conditions issues. CORRECT
- Cross-functions race conditions issues. CORRECT
- Overflow and underflow issues. CORRECT
- Logic of the economy model. CORRECT
- Exchange rate impact. CORRECT
- Oracle calls. CORRECT
- Timestamp dependace. CORRECT
- Potential delays in data delivery. CORRECT
- Permissions on execution methods. CORRECT
- Private data leakage prevention. CORRECT

Audit findings

With MythX, each smart contract is compiled individually and checked for a range of security issues using static and dynamic analysis. The following table lists the bug classes that were tested for. A checkmark in the "pass" column indicates that no issues were detected in the category. An "X" indicates that one or more issues in the category were found.

SWC ID	Bug class	Pass
SWC-100	Function Default Visibility	✔
SWC-101	Integer Overflow and Underflow	✔
SWC-102	Outdated Compiler Version	✖
SWC-103	Floating Pragma	✔
SWC-104	Unchecked Call Return Value	✔
SWC-105	Unprotected Withdrawal	✔
SWC-106	Unprotected SELFDESTRUCT Instruction	✔
SWC-107	Reentrancy	✔
SWC-108	State Variable Default Visibility	✔
SWC-109	Uninitialized Storage Pointer	✔
SWC-110	Assert Violation	✔
SWC-111	Use of Deprecated Solidity Functions	✔
SWC-112	Delegatecall to Untrusted Callee	✔
SWC-113	DoS with Failed Call	✔
SWC-114	Transaction Order Dependence	✔
SWC-115	Authorization through tx.origin	✔
SWC-116	Timestamp Dependence	✔
SWC-118	Incorrect Constructor Name	✔
SWC-119	Shadowing State Variables	✔

SWC-120	Weak Sources of Randomness	✔
SWC-123	Requirement Violation	✔
SWC-124	Write to Arbitrary Storage Location	✔
SWC-127	Arbitrary Jump	✔
SWC-128	Gas Exhaustion	✔
SWC-129	Typographical Error	✔
SWC-130	Right-To-Left-Override control character	✔

High Severity Findings

- No high severity issues found.

Medium Severity Findings

- No medium severity issues found.

Low Severity Findings and Recommendations

- No low severity issues found. Two recommendations.

1. **Zero address checking**

Recommendation:
Add zero address checking functions. Developer has added before deployment of contracts

2. **Prefer External to public function declaration**

Recommendation:
Use the external visibility modifier for functions never called from the contract via

internal call. Functions to be update to external functions.

Compiler version is a recurrent informational finding in these contracts. After discussion with developer, he has stated that the compiler version used for the development is not the last one, because the previous version is battle tested and due to this, he can apply better coding and avoid potential unknown issues to the newer version.

The contracts are compiling propertly and on top of this, developer has agreed to allow compiler versions above the version he has used at coding time.

There is another finding not related to code but related with the Gnosis multi signarute system usage. It is documented that main protocols will follow a 6/9 model.

The governance multisig is controlled by 9 community members and requires 6 signatures. This means that no transactions can be processed until the 9 signers have reached at least 66.67% consensus.

But at the moment of this audit, the Gnosis system is working under a 3/5 model due to a technical limitation in the Gnosis systems, which avoid the community to increase the number of signatures from 5 to 9.

After confirmation with developers, they are aware of this and they already have an open ticket with Binance for the resolution of this limitation. This point will be updated in following versions of this document.

Conclusion

At the moment of conducting these audits, Ballena.io smart contracts do not contain any known security issue. All minor issues and recommendations have been solved or implemented before live version.

On top of this, code and functionalities have been deeply tested by the community users, following a testing plan, before the final deployment of contracts.

Ballena.io application code is secure and do not have any security issue before live version.