STAYSAFU AUDIT

August 18TH, 2022

Melega

TABLE OF CONTENTS

AUDIT SUMMARY

This report was written for Melega ($Marco) in order to find flaws and vulnerabilities in the Melega project's source code, as well as any contract dependencies that weren't part of an officially recognized library.

A comprehensive examination has been performed, utilizing Static Analysis, Manual Review, and Melega Deployment techniques. The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors
- Assessing the codebase to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
- Through line-by-line manual review of the entire codebase by industry expert

AUDIT OVERVIEW

PROJECT SUMMARY

Project name	Melega
Description	Melega (Marco) is the flagship utility token of melega.finance, the best AMM DEX on Binance Smart Chain (BSC) providing friendly trading and better project support. A total black design, an original name, a distinctive and recognizable logo.
Platform	BNB Chain
Language	Solidity
Codebase	https://bscscan.com/token/0x963556de0eb8138e97a85f0a86ee0acd159d210b

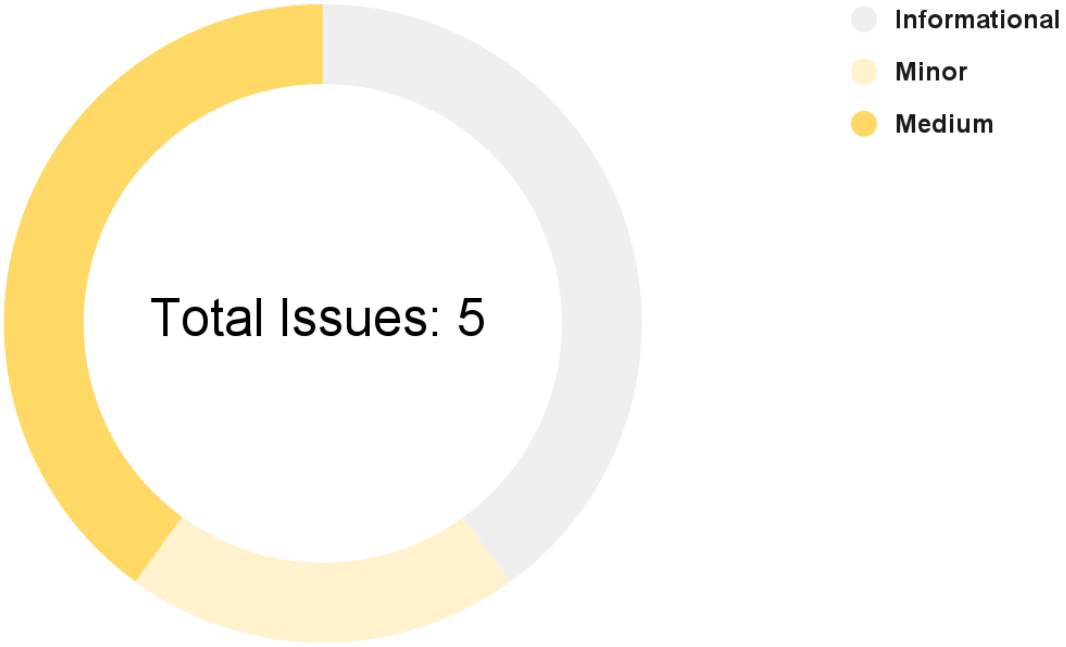
FINDINGS SUMMARY

Vulnerability	Total
Critical	0
Major	0
Medium	2
Minor	1
Informational	2

EXECUTIVE SUMMARY

There have been no major or critical issues related to the codebase and all findings listed here range from informational to medium. The medium security issues are the dependence on a decentralized exchange platform and centralization of privileges.

AUDIT FINDINGS



Code	Title	Severity
CENT-3	Centralization of initial token distribution	Medium
EXT-1	External protocol dependencies	Medium
GAS-3	Unoptimized function type	Minor
FUNC-1	Unused functions	Informational
COMP-1	Unlocked compiler version	Informational

CENT-3 | Centralization of initial token distribution

Description

A constructor (line 469) within the contract mints 100% of the initial token supply to the deployer address (msg.sender). This initially centralizes token supply to the deployer address.

Recommendation

We recommend decentralizing tokens as soon as possible, matching the project's intentions. Examples of this are burning tokens or adding tokens to a liquidity pool (locked). We also recommend being fully transparent with the community about token distribution.

EXT-1 | Dependence to external protocol

Description

The contract interacts with PancakeSwap protocols. The scope of the audit would treat these third party entities as black boxes and assume they are fully functional. However in the real world, third parties may be compromised thus leading assets to be lost or stolen. We fully understand that the business logic of the Melega project is designed to work with PancakeSwap protocols. This extends to other protocols and interfaces not within the scope of this audit.

Recommendation

We encourage the team to constantly monitor the security level of the entirety of PancakeSwap protocols interacted with, as the security of the project is highly dependent on the security of these decentralized exchange platforms.

GAS-3 | Unoptimized function type

Description

Throughout Melega's contracts some functions are of type public although they are never called within the contract. External functions require significantly less gas to call. Such found functions are listed below:

- decreaseAllowance -> Line 657
- increaseAllowance -> Line 630
- transferFrom -> Line 601
- approve -> Line 578
- allowance -> Line 561
- transfer -> Line 548
- balanceOf -> Line 530
- totalSupply -> Line 523
- decimals -> Line 516
- symbol -> Line 499
- name -> Line 491
- transferOwnership -> Line 183
- renounceOwnership -> Line 175

Recommendation

We recommend reviewing each of the functions listed above and where possible switch their type from public to external.

FUNC-1 | Unused functions

Description

Multiple functions within Melega's contract are defined as private or internal but are never called within the contract. This wastes contract space as there is a maximum size a contract can have. Functions found with this issue have been listed below:

- _setupDecimals -> Line 780
- _burn -> Line 735
- trySub -> Line 234
- tryMul -> Line 246
- tryMod -> Line 275
- tryDiv -> Line 263
- tryAdd -> Line 221
- sub -> Line 306
- mul -> Line 320
- mod -> Line 416
- mod -> Line 350
- div -> Line 390
- div -> Line 334
- _msgData -> Line 115

Recommendation

We recommend removing these functions from the contract.

COMP-1 | Unlocked compiler version

Description

Melega's contract does not have locked compiler versions, meaning a range of compiler versions can be used. This can lead to differing bytecodes being produced depending on the compiler version, which can create confusion when debugging, as bugs may be specific to a specific compiler version(s).

Recommendation

To rectify this, we recommend setting the compiler to a single version, the version tested the most to be compatible with the code, an example of this change can be seen below.

```
pragma solidity 0.8.7;
```

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement.

This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement.

This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without StaySAFU'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 considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts StaySAFU to perform a security assessment.

This report 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 in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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.

StaySAFU's position is that each company and individual are responsible for their own due diligence and continuous security. StaySAFU'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 fun.