AUDIT

# CryptoHealth

CoinMooner

Find Next Moonshot coins

# TABLE OF CONTENTS

# AUDIT SUMMARY

This report was written for CryptoHealth (CHT) in order to find flaws and vulnerabilities in the CryptoHealth 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 CryptoHealth 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 a line-by-line manual review of the entire codebase by industry expert

# AUDIT OVERVIEW

## PROJECT SUMMARY

| | |
|---|---|
| Project name | CryptoHealth |
| Description | Project that aims to redefine HODLing by building a single payment health insurance policy for both crypto traders and non-traders, and providing real-world and unique utilities thru a portfolio of tokens |
| Platform | BNB Smart Chain |
| Language | Solidity |
| Codebase | https://bscscan.com/address/0x57ee76aee8b02c1e3cea9cf2046f741b81916613 |

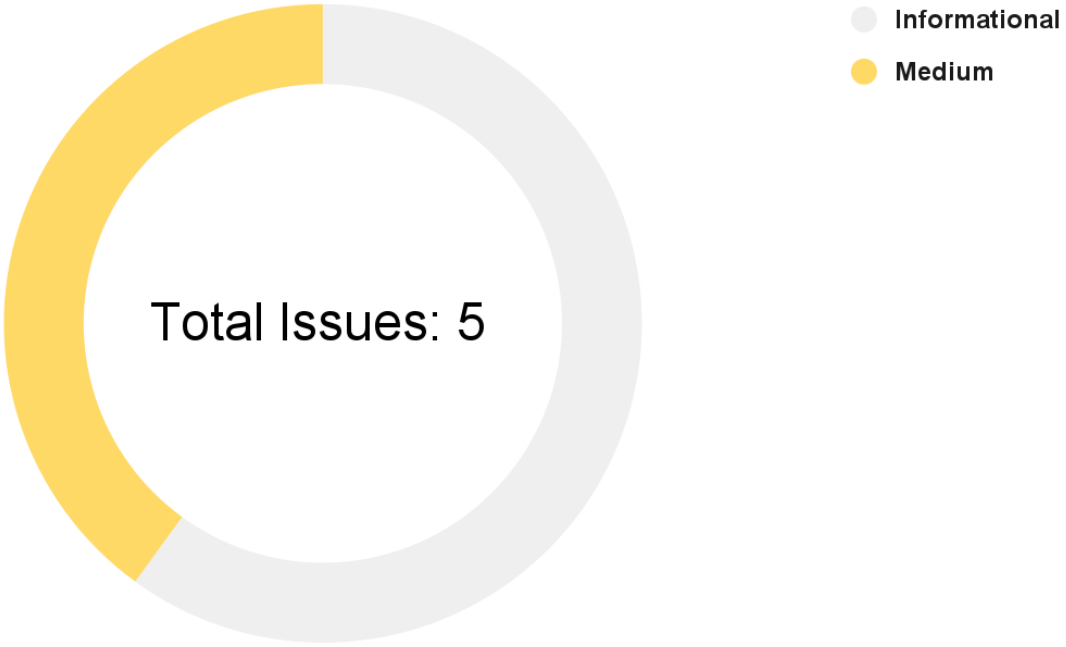
## FINDINGS SUMMARY

| Vulnerability | Total |
|---|---|
| Critical | 0 |
| Major | 0 |
| Medium | 2 |
| Minor | 0 |
| Informational | 3 |

# 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: centralization of privileges and dependence on external protocols.

# AUDIT FINDINGS




| Code | Title | Severity |
|---|---|---|
| CENT-1 | Centralization of major privileges | Medium |
| EXT-1 | Dependence to external protocol | Medium |
| COMP-1 | Unlocked compiler versions | Informational |
| FUNC-1 | Unused functions | Informational |
| GAS-3 | Unoptimized function type | Informational |

# CENT-1 | Centralization of major privileges

## Description

The onlyOwner modifier of the smart contract(s) gives major privileges over it (Change fees)*. This can be a problem, in the case of a hack, an attacker who has taken possession of this privileged account could damage the project and the investors.
*This list is not exhaustive but presents the most sensitive points

## Recommendation

We recommend at least using a multi-sig wallet as the owner address, and at best to establish a community governance protocol to avoid such centralization. For more information, see https://solidity-by-example.org/app/multi-sig-wallet/

## 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 Crypto Health 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.

## COMP-1 | Unlocked compiler version

### Description

CryptoHealth'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 is compatible with the code, an example of this change can be seen below.

```
pragma solidity 0.8.0;
```

## FUNC-1 | Unused functions

### Description

Multiple functions within CryptoHealth'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:

- trySub -> Line 328
- tryMul -> Line 335
- tryMod -> Line 345
- tryAdd -> Line 320
- mod -> Line 394
- mod -> Line 368
- div -> Line 383
- _msgData -> Line 181
- verifyCallResult -> Line 297
- sendValue -> Line 227
- isContract -> Line 222
- functionStaticCall -> Line 271
- functionStaticCall -> Line 267
- functionDelegateCall -> Line 286
- functionDelegateCall -> Line 282
- functionCallWithValue -> Line 254

- functionCallWithValue -> Line 246
- functionCall -> Line 238
- functionCall -> Line 234

## Recommendation

We recommend safely removing these functions from the contract.

## GAS-3 | Unoptimized function type

### Description

Throughout CryptoHealth'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:

- checkLastTransfer -> Line 775
- excludeFromReward -> Line 721
- reflectionFromToken -> Line 693
- deliver -> Line 681
- isSniperAccount -> Line 677
- isExcludedFromFees -> Line 673
- isExcludedFromReward -> Line 669
- decreaseAllowance -> Line 653
- increaseAllowance -> Line 640
- transferFrom -> Line 623
- approve -> Line 614
- allowance -> Line 605
- transfer -> Line 596
- totalSupply -> Line 587

- decimals -> Line 582
- symbol -> Line 578
- name -> Line 574
- transferOwnership -> Line 208
- renounceOwnership -> Line 204

## Recommendation

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

# Global security warnings

These are safety issues for the whole project. They are not necessarily critical problems but they are inherent in the structure of the project itself. Potential attack vectors for these security problems should be monitored.

## CENT-1 | Global SPOF (Single Point Of Failure)

The project's smart contract has a problem of centralized privileges. The owner system in particular can be subject to attack. To address this security issue we recommend using a multi-sig wallet, establishing secure project administration protocols, and strengthening the security of project administrators.

# 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 CoinMooner'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 CoinMooner 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.

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