STAYSAFU AUDIT

July 18TH, 2022

BitBox

TABLE OF CONTENTS

AUDIT SUMMARY

This report was written for BitBox ($BBX) in order to find flaws and vulnerabilities in the BitBox 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 BitBox 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	BitBox
Description	BitBox is a project created to serve in real and virtual environments, social activities and commercial areas. Wait to be caught in the middle of reality and digitalized choices… [BitBox: Team]
Platform	Binance Smartchain
Language	Solidity
Codebase	https://bscscan.com/address/0x353Ce5173D4C1f08Ebae5BEe7583e310067c5415#code

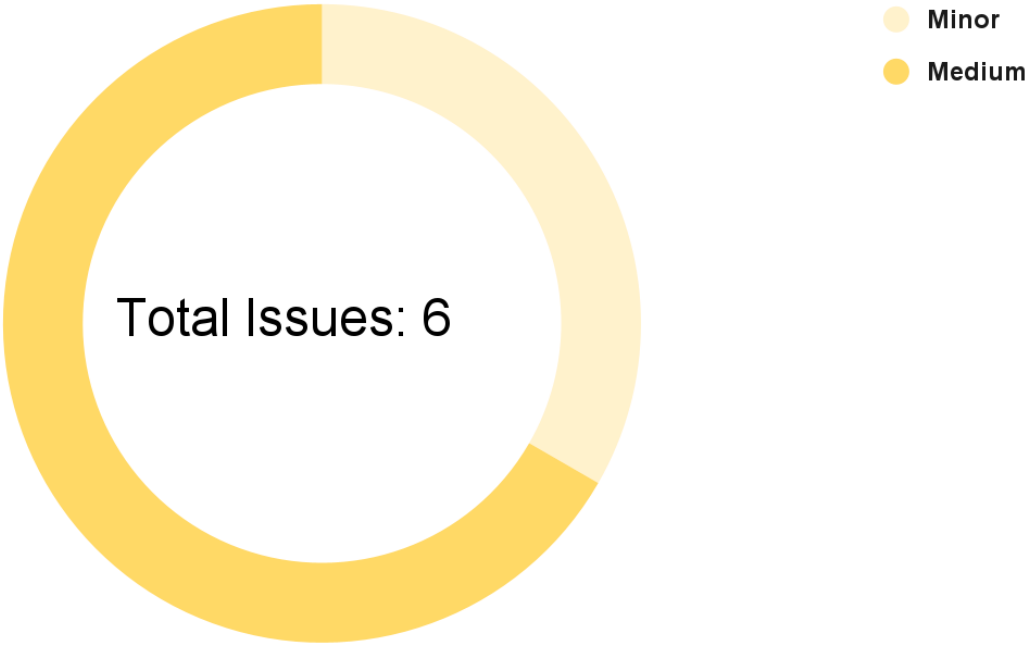
FINDINGS SUMMARY

Vulnerability	Total
Critical	0
Major	0
Medium	4
Minor	2
Informational	0

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, centralization of privileges, and missing threshold checks.

AUDIT FINDINGS



Code	Title	Severity
CENT-1	Centralization of major privileges	Medium
EXT-1	External protocol dependencies	Medium
THRE-1	Missing threshold checks	Medium
TX-1	Use of tx.origin in authorisation	Medium
COMP-1	Unfixed version of compiler	Minor
BLOC-1	Use of block.timestamp	Minor

CENT-1 | Centralization of major privileges

Description

The onlyOwner modifier in the smart contract(s) give major privileges over them (blacklists, updating fees, change ownership, and pausing the ability to make transfers (buy/sell))*. 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 to use a multi-sig wallet as the owner address, and at best to establish a community governance protocol to avoid such centralization.

See: https://solidity-by-example.org /app/multi-sig-wallet/

EXT-1 | Dependence to an external protocol

Description

The contract serves as an underlying entity to interact with third party PancakeSwap protocols. The scope of the audit would treat this third party entity as black box and assume it is fully functional. However in the real world, third parties may be compromised and may have led to assets lost or stolen.

Recommendation

We encourage the team to constantly monitor the security level of the entire PancakeSwap project, as the security of the token is highly dependent on the security of the decentralized exchange platform.

THRE-1 | Missing threshold checks

Description

Functions which can change sensitive variables within BitBox's contract do not contain threshold checks to ensure these variables are not changed to unreasonable values. This includes fees. As such it is important to add a threshold to prevent an attacker from setting fees as 100% easily. Key examples of Identified functions with this issue have been listed below:

- setBuyTax -> Line 1082
- setSellTax -> Line 1092

Recommendation

We recommend adding threshold checks using require statements for each of the identified functions above and other functions with this issue.

TX-1 | Use of tx.origin in authorisation

Description

The use of tx.origin is strongly deprecated by the industry. It can lead to phishing attacks by falsifying the identity of the original caller of the function. This is especially problematic as tx.origin is used within the onlyOwner modifier (line 511), thus this issue extends to all functions which use this modifier. Read more about it here.

Recommendation

In the case of this smart contract, we recommend using msg.sender instead of tx.origin.

Newer versions of Ownable.sol (from OpenZeppilin) do not use tx.origin. As such we recommend upgrading to the latest Ownable.sol iteration.

BLOC-1 | Use of block.timestamp

Description

The use of block.timestamp can be problematic. The timestamp can be partially manipulated by the miner (see https://cryptomarketpool.com/block-timestamp-manipulation-attack/).

Recommendation

We fully understand that the use of block.timestamp within the BitBox Protocol is required for certain functionality such as the adding liquidity. Nevertheless, it is still useful to point out this kind of potential security problem.

COMP-1 | Unfixed version of compiler

Description

BitBox token'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 0.8.9;
```

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.