STAYSAFU
AUDIT

September 27th, 2022

PiConnect

TABLE OF CONTENTS

AUDIT SUMMARY

This report was written for PiConnect in order to find flaws and vulnerabilities in the PiConnect project's source code, as well as any contract dependencies that were not part of an officially recognized library.

A comprehensive examination has been performed, utilizing Static Analysis, Manual Review, and PiConnect 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	PiConnect
Description	Pi Connect aims to solve the existing limitation on Pi Network crypto
Platform	BNB Smart Chain
Language	Solidity
Codebase	https://bscscan.com/token/0x6e1d1f8f91e5c9c35b8fd361471286487cc1eaa4

FINDINGS SUMMARY

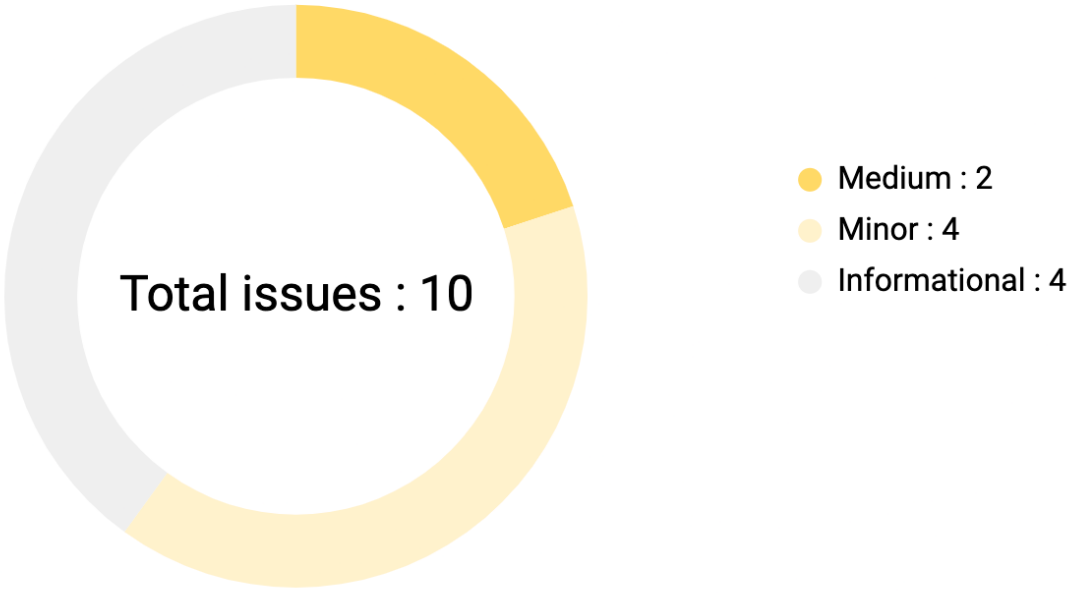
Vulnerability	Total
Critical	0
Major	0
Medium	2
Minor	4
Informational	4

EXECUTIVE SUMMARY

Pi Connect aims to solve the existing limitation on Pi Network crypto, including mining rewards and liquidity provision. Furthermore, the application provides users with information about the price and exchange rate between Pi for other mainstream. Aims to solve the existing limitation on Pi Network crypto, including mining rewards and liquidity provision. Furthermore, the application provides users with information about the price and exchange rate between Pi for other mainstream.

There have been no major or critical issues related to the codebase and all findings listed here are minor or informational. The major issues that have been found are centralization of major privileges and dependence on external protocols.

AUDIT FINDINGS



Code	Title	Severity
CENT-1	Centralization of major privileges	Medium
EXT-1	External protocol dependencies	Medium
COMP-1	Unfixed version of compiler	Minor
CDI-1	Contract doesn't import packages from Verified sources	Minor
MAE-1	Missing Arithmetic Events	Minor
MZC-1	Missing Zero Check	Minor
CBC-1	Comparison with Boolean Constants	Informational

SLV-1	Shadowing Local Variables	• Informational
DEA-1	Dead Code	• Informational
CSV-1	Constable State Variables	• Informational

CENT-1 | Centralization of major privileges

Description

The onlyOwner modifier of the smart contract gives major privileges over it. The owner can update/manipulate the following in the contract:

- Send locked tokens to an address but only if the address is not a private sale holder
- Include/Exclude wallets/accounts from fee
- Set ICO Date and can change it any time, even after the start of the ICO. Thus, the owner can restart the ICO
- Set tax but not more than 15%
- Enable/Disable tax on transactions

An attacker can also manipulate these parameters for own gains if the owner's private keys are compromised.

**This list is not exhaustive but presents the most sensitive points*

Recommendation

We recommend at least to use a multi-sig wallet as 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 an external protocol

Description

The contract is serving as the underlying entity to interact with third party Uniswap 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 lead to lost or stolen assets.

Recommendation

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

COMP-1 | Unfixed version of compiler

Description

PiConnect'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).

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

Recommendation

We recommend fixing the compiler version to the most recent one:

```
//l 6
pragma solidity 0.8.16;
```

CDI-1 | Contract doesn’t import packages from Verified sources

Description

PiConnect is a flattened contract and does not directly imports packages from verified sources like OpenZeppelin etc.

Recommendation

We recommend importing all packages from npm directly without flattening the contract. Functions could be modified or can be susceptible to vulnerabilities.

MAE-1 | Missing Arithmetic Events

Description

Events are missing from critical parameter changes in the contract on lines -

#1335, #1331, #1342, #1353, #1356

Recommendation

We recommend to emit events when there are critical parameter changes in the contract for better transparency and to keep track of the changes.

MZC-1 | Missing Zero Check

Description

Missing zero address validation from these functions may lead to setting of critical parameter's values as zero address.

Effected Lines: #1353

Recommendation

We recommend to add a check to verify that the address/'s passed in these functions is/are not zero.

CBC-1 | Comparison to Boolean Constants

Description

Comparison to Boolean constants is not required in solidity because they can be used directly without the need to compare with “true or false”

Effected Lines:

- #1369,
- #1387,
- #1369,
- #1357,
- #1087,
- #1090,
- #1093,
- #1145,
- #1148,
- #1151

Recommendation

We recommend removing the direct comparisons as it is not required.

SLV-1 | Shadowing Local Variables

Description

PiConnect shadows local variables from the “Ownable” contract on lines L#1302 and L#1103

Recommendation

Rename the local variables that shadow other component

DEA-1 | Dead Code

Description

Many functions are not used in the contract and should be removed because it makes the code's review more difficult and affect the code's readability negatively.

Effected lines –

- #440 to #487,
- #873,
- #1319,
- #817,
- #628 to #698,
- #777,
- #817,
- #843

Recommendation

Remove all the Dead Code or Unused functions.

CSV-1 | Constable State Variables

Description

Many state variables are never updated in the contract and should be declared constant for better optimization

Effected lines –

- #987 to #989
- #993
- #997 to #1003
- #1005

Recommendation

Remove all the Dead Code or Unused functions.

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.

The Company only to the extent permitted under the terms shall use this report provided in connection with the Services set forth in the Agreement 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.