PROOF

PIG FINANCE

SMART CONTRACT

SECURITY AUDIT REPORT

Disclaimer

This is a limited report of findings based on an analysis of industry best practices as of the date of this report regarding cybersecurity vulnerabilities and issues in smart contract frameworks and algorithms, the details of which are detailed in this report. stated in the report. To get the full picture of our analysis, it's important to read the full report. Although we have conducted our analysis and have done our best to prepare this report, you should not rely on this report and cannot claim against us based on what it does or does not say or how it was produced. It is important to do your own research before making any decisions. This is explained in more detail in the following disclaimer. Please be sure to read to the end.

Disclaimer:

BY READING THIS REPORT OR ANY PART THEREOF, YOU AGREE TO THE TERMS OF THIS DISCLAIMER. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Proof Audit and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (Proof Audit) owe no duty of care towards you or any other person, nor does Proof Audit make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and Proof Audit hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, Proof Audit hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against Proof Audit, for any amount or type of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

Security analytics are based solely on smart contracts. Application or process security not checked. Product code not reviewed.

Table of Contents

Executive Summary

Objectives

Proof Audit, carried out an audit of PIG FINANCE, specifically their BEP20 token. The project is based on the BNB Chain Network. We reviewed documentation which helped with understanding the functions of their code. Our findings in the audit ranged from minor to critical.

Project Info

Audited project

PIG FINANCE

Deployer Address

0x54901D18fc35Eb2AA8fA7017d6b8a651c4390191

Contract Address

0x8850D2c68c632E3B258e612abAA8FadA7E6958E5

Blockchain

BNB Chain

Project website:

https://www.pigtoken.finance/

Methodology

During the audit process, we inspected the repository thoroughly, using a line-by-line code read through to review vulnerabilities, quality of the code and adherence to best practices and specifications. We used Computer-Aided Verification to support the audit process.

Our auditing process is as follows:

1. **Code Review:**
   A review of the scope, specifications and documentation provided to ensure an in depth understanding of the purpose and functionality of the relevant smart contracts.

2. **Automated Analysis:**
   A series of reviews carried out with the use of automated tools. These reviews serve as a basis for further manual analysis and provide relevant visualizations of the code.

3. **Testing & Manual Review of Code:**
   Test coverage analysis and a line-by-line read through of the project code in order to identify vulnerabilities, errors and weaknesses in code quality.

4. **Specification Comparison:**
   A review of the code against the specifications provided to ensure that the code operates as is intended.

5. **Best Practices Review:**
   A review of the smart contracts to identify potential improvements in effectiveness, efficiency, and maintainability, with a focus on adherence to industry best practices.

PROOF

Scope

The contracts audited are from the pigtokenfinance/contracts git repository. The audit is based on the commit 'Create contract.sol' from 08/03/2021.

The audited contracts are:

contract.sol

The scope of the audit is limited to these files. No other files in this repository were audited. Its dependencies are assumed to work according to their documentation. Also, no tests were reviewed for this audit.

### PIG FINANCEv3.sol Interaction Graph

Analyses

Without being limited to them, the audit process included the following analyses:

- Arithmetic errors
- Outdated version of Solidity compiler
- Race conditions
- Reentrancy attacks
- Misuse of block timestamps
- Denial of service attacks
- Excessive gas usage
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures
- Centralization and upgradeability

Summary of Findings

We found **0** critical issue, **2** Major issues, **1** medium issues, and **3** minor issues.

Findings

■ Critical   ■ Major   ■ Medium   ■ Minor
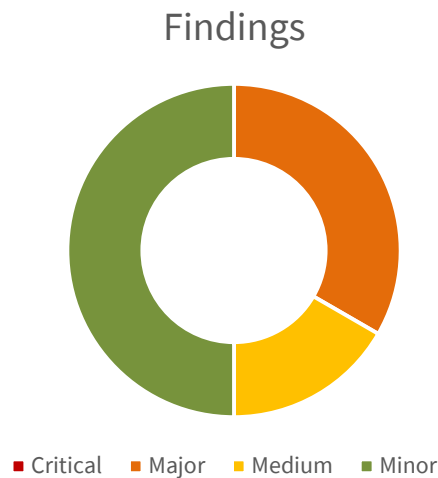
Security Issues

ID	Title	Severity	Status
01	Centralized Risk in addLiquidity()	Major	**Acknowledged**
02	Centralization Risk	Major	**Acknowledged**
03	Contract gains non-withdrawable BNB via the swapAndLiquify() function	Medium	**Acknowledged**
04	Possible To Gain Ownership After Renouncing the Contract Ownership	Minor	**Acknowledged**
05	Third Party Dependencies	Minor	**Acknowledged**
06	Lack of Input Validation	Minor	**Acknowledged**

Findings

Severity Classification

Security risks are classified as follows:

- **Critical:** These are issues that we manage to exploit. They compromise the system seriously. They must be fixed **immediately**.
- **Medium:** These are potentially exploitable issues. Even though we did not manage to exploit them, or their impact is not clear, they might represent a security risk in the near future. We suggest fixing them **as soon as possible**.
- **Minor:** These issues represent problems that are relatively small or difficult to take advantage of but can be exploited in combination with other issues. These kinds of issues do not block deployments in production environments. They should be taken into account and be fixed **when possible**.

Issues Status

An issue detected by this audit can have four distinct statuses:

- **Unresolved**: The issue has not been resolved.
- **Acknowledged**: The issue remains in the code but is a result of an intentional decision.
- **Resolved**: Adjusted program implementation to eliminate the risk.
- **Partially resolved**: Adjusted program implementation to eliminate part of the risk. The other part remains in the code but is a result of an intentional decision.
- **Mitigated**: Implemented actions to minimize the impact or likelihood of the risk.

Critical Severity Issues

N/A

PROOF

Major Severity Issues

Centralized Risk In `addLiquidity`

Description: The addLiquidity function calls the `uniswapV2Router.addLiquidityETH` function with the to address specified as `owner()` for acquiring the generated LP tokens from the PIG-BNB pool. As a result, over time the _owner address will accumulate a significant portion of LP tokens. If the _owner is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Recommendation: We advise the to address of the `uniswapV2Router.addLiquidityETH` function call to be replaced by the contract itself, i.e. `address(this)` , and to restrict the management of the LP tokens within the scope of the contract's business logic.

Status: **Acknowledged**

Centralization Risk

Description: The contract owner has the capability to:
- Owner has the privilege to exclude from reward through `excludeFromReward()`
- Owner has the privilege to include from reward through `includeInReward()`
- Owner has the privilege to exclude from fee through `excludeFromFee()`
- Owner has the privilege to include from fee through `includeInFee()`
- Owner has the privilege to set tax fee percent through `setTaxFeePercent()`
- Owner has the privilege to set liquidity fee percent through `setLiquidityFeePercent()`
- Owner has the privilege to lock the contract through `lock()`
- Owner has the privilege to enable `swapAndLiquifyEnabled` through `setSwapAndLiquifyEnabled()`
- Owner has the privilege to set `_maxTxAmount` through `setMaxTxPercent()`

Recommendation: We advise the client to carefully manage the owner account's private key to avoid any potential risks ofbeing hacked.

Status: **Acknowledged**

Medium Severity Issues

Contract gains non-withdrawable BNB via the swapAndLiquify() function

Description: The swapAndLiquify function converts half of the contractTokenBalance PIG tokens to BNB. The other half of PIG tokens and part of the converted BNB are deposited into the PIG-BNB pool on pancakeswap as liquidity. For every swapAndLiquify function call, a small amount of BNB is leftover in the contract.

Recommendation: We recommend adding a withdraw function to allow for withdrawing BNB

Status: **Acknowledged**

Minor Severity Issues

Possible To Gain Ownership After Renouncing the Contract Ownership

Description: AcknowledgedDescriptionAn owner has the possibility to gain ownership of the contract even if he calls function renounceOwnershipto renounce the ownership.

Recommendation: We advise updating/removing lock and unlock functions in the contract.

Status: **Acknowledged**

Third Party Dependencies

Description: The contract is serving as the underlying entity to interact with third-party PancakeSwap protocols. The scope of the audit would treat those 3rd party entities as black boxes and assume their functional correctness. However, in the real world, 3rd parties may be compromised and lead to assets being lost or stolen.

Recommendation: PTP-05 | Third Party DependenciesCategory Severity Location StatusControl Flow Minor PigToken.sol: 744 AcknowledgedDescriptionThe contract is serving as the underlying entity to interact with third-party PancakeSwap protocols. Thescope of the audit would treat those 3rd party entities as black boxes and assume their functionalcorrectness. However, in the real world, 3rd parties may be compromised and lead to assets being lost orstolen.Recommendation

Status: **Acknowledged**

Lack of Input Validation

Description: The variables _taxFee and _liquidityFee should not exceed 100 respectively, and the sum of them should not exceed 100. The variable maxTxPercent should be less than or equal to 100 and greater than 0.

Recommendation: We advise the client to check that the variables _taxFee , _liquidityFee and maxTxPercent like as follows

Status: **Acknowledged**

Security Rating

PROOF

84%
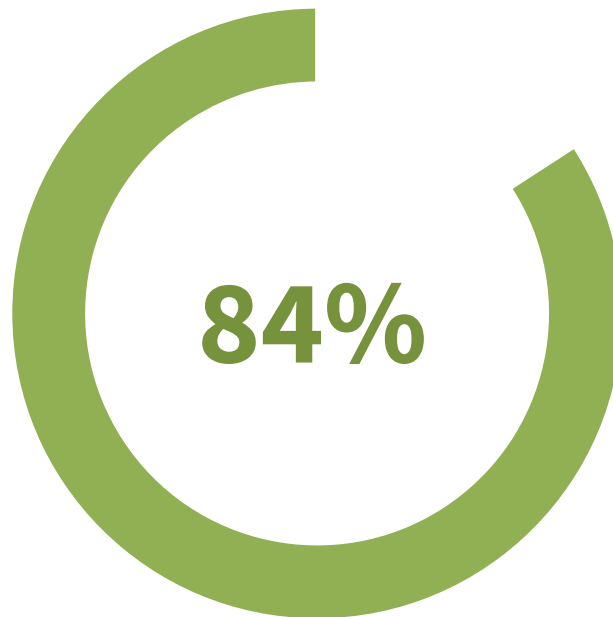
Based on Vulnerabilities Found

PROOF