SMART CONTRACT REVISION AUDIT PUBLIC REPORT

Revision Audit of Folks Finance's Algorand Smart Contracts - Changes to Token Pair and Oracle Adapters

VPQ-20220215

BlockchainItalia.io - Folks Finance

2022-05-12

VANTAGEPOINT
Security at the Speed of Development

TABLE OF CONTENTS

1. Executive Summary

Overview

Vantage Point Security Pte Ltd was engaged by BlockchainItalia.io to conduct an Algorand smart contract security revision audit of changes to Folks.Fianance Token Pair and Oracle Adapter smart contracts which are part of the update to support Liquidity Pool Tokens within the Folks.Finance Protocol to identify security vulnerabilities, weaknesses and any instances of non-compliance to best practices within the smart contract.

Algorand smart contract security review was conducted based on the following materials provided.

- Supporting Documents
    - Supporting documents for audit containing definitions and transaction details
        - LP_Tokens_Audit.pdf
    - Internal Study Report on Impact of LP Tokens
        - LPTOKENREPORT.pdf
- PyTeal Code
    - Private Repo
        - https://github.com/blockchain-italia/ff-vp-contracts/commits/master Commit ID 2eda279e44c0dd2f15ccadba14e4be48d76c6844

Vantage Point performed this review by first understanding the changes to Folks.Finance protocol in view of support for Liquidity Pool tokens and PyTeal code for the affected contracts. We sought clarifications on potential issues, discrepancies, and flaws within the smart contract's logic through discussions with the Folks.Finance team.

A revision audit was conducted on the provided PyTeal code to identify any weaknesses, vulnerabilities, and non-compliance to Algorand best practices from the new changes introduced to the code. Test cases included in this review have been amended in the appendix of this document for completeness.

The following informational issues were noted from this review.

1. **Considerations for Global Borrow Limit**
   Global Borrow Limit is introduced to limit the risk and potential damages to Folks.Finance protocol in an unexpected event. However, as the token pair contract is used to impose the global state *total_borrowed_limit* based on the ASA ID of the collateral asset and the borrow asset, Folks.Finance team should consider AMMs such as Pact.Fi could have 4 different ASA IDs for the same LP pair due to 4 different fee tiers. As long as the total_borrowed_limit is intended to limit the size of the loans that can be created for a single LP token from a single AMM, Folks.Finance team should consider this in registering LP tokens as collaterals and setting global borrow limits for such.
2. **Insufficient On-Chain Validations Against Altered Transactions**
   Often, Algorand smart contracts make use of front-end web applications to forward transactions groups which are eventually signed by users, having on-chain validations help when the front-end web applications have been compromised, as a layer of defence. However, the highlighted instances in the current report are either only allowed for administrators or

require other transaction groups to be included with additional validations and therefore have significantly less concerns in terms of security and thus, the recommended remediation is to state details regarding the lack of group size checks in the documentation for clarity.

The outcome of this Algorand Smart Contract Security Review engagement is provided as a detailed technical report that provides the Smart Contract owners a full description of the vulnerabilities identified, the associated risk rating for each vulnerability, and detailed recommendations that will resolve the identified technical issue.

VULNERABILITY OVERVIEW

Severity	Count	Open	Closed
Critical	**0**	**0**	**0**
High	**0**	**0**	**0**
Medium	**0**	**0**	**0**
Low	**0**	**0**	**0**
Observational	**2**	**0**	**2**
Summary	**2**	**0**	**2**

Vulnerability Risk Score

All vulnerabilities found by Vantage Point will receive and individual risk rating based on the following four categories.

CRITICAL COMPONENT RISK SCORE

Critical severity findings relate to an issue, which requires immediate attention and should be given the highest priority by the business as it will critically impact business interest critically.

HIGH COMPONENT RISK SCORE

HIGH severity findings relate to an issue, which requires immediate attention and should be given the highest priority by the business.

MEDIUM COMPONENT RISK SCORE

A MEDIUM severity finding relates to an issue, which has the potential to present a serious risk to the business.

LOW COMPONENT RISK SCORE

LOW severity findings contradict security best practice and have minimal impact on the project or business.

OBSERVATIONAL

Observational findings relate primarily to non-compliance issues, security best practices or are considered an additional security feature that would increase the security stance of the environment which could be considered in the future versions of smart contract.

2. Project Details

Scope

Contact Name	Gidon Katten
Contact Email	gidonkatten@blockchainitalia.io
Application Name	Folks.Finance Changes to Token Pair and Oracle Contract
GIT Commit ID	2eda279e44c0dd2f15ccadba14e4be48d76c6844
Items Completed	Items Completed • assets/oracle_adapter_2_approval_program.py • assets/oracle_adapter_2_clear_program.py • assets/token_pair_approval_program.py

Component	Review Type	Status
Algorand Smart Contract	Smart Contract Security Review	Completed
Algorand Smart Contract	Smart Contract Security Review Retest	Completed

VERSION HISTORY

Date	Version	Release Name
12th May 2022	V0.1	Draft
13th May 2022	V0.2	QA Release
13th May 2022	V1.0	Final

3. Risk Assessment

This chapter contains an overview of the vulnerabilities discovered during the project. The vulnerabilities are sorted based on the risk categories of CRITICAL, HIGH, MEDIUM and LOW. The category OBSERVATIONAL refers to vulnerabilities that have no risk score and therefore have no immediate impact on the system.

Overview of Components and Their Vulnerabilities

1. Revision Audit of Folks Finance's Algorand Smart Contracts - Changes to Token Pair and Oracle Adapters		OBSERVATIONAL	ⓘ
1.1. Considerations for Global Borrow Limit	**Closed**	**OBSERVATIONAL**	ⓘ
1.2. Insufficient On-Chain Validation Against Altered Transactions	**Closed**	**OBSERVATIONAL**	ⓘ

4. DETAILED DESCRIPTION OF VULNERABILITIES

2. REVISION AUDIT OF FOLKS FINANCE'S ALGORAND SMART CONTRACTS - CHANGES TO TOKEN PAIR AND ORACLE ADAPTERS

OBSERVATIONAL

1.1. Considerations for Global Borrow Limit

OBSERVATIONAL

VULNERABILITY TRACKING

STATUS: **Closed**

BACKGROUND

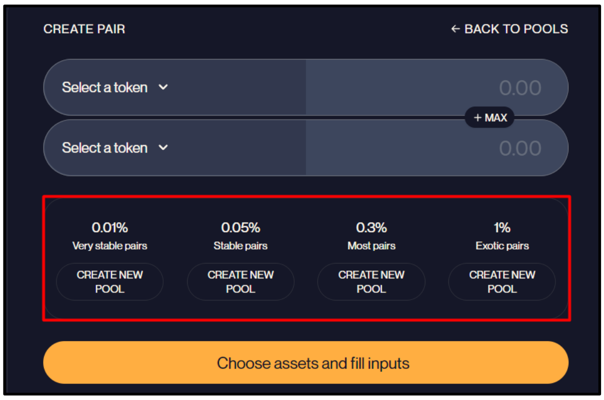
LP Tokens from Tinyman and Pact.Fi can be used as collateral for loans from Folks.Finance is based on a safe pricing formula which retrieves a fair value of the LP token even when there are actors with malicious intent to skew a concerned LP's reserves or pricing via swapping or donating large amount of assets. For security reasons, global borrow limits are imposed for each token pair.

DESCRIPTION

Instance 1 - Global Borrow Limit - Multiple LP Tokens with Identical Asset Combination

Currently, a global borrow limit can be set for a pair of collateral assets (Example - Tinyman LP USDC-Algo) and a borrow asset (Example - goBTC). If the intention is to limit the amount of risk and damage per each collateral and borrow pair, it is important to note that AMMs such as Pact.Fi have different fee tiers and therefore separate LP tokens for the same combination of assets.

Screenshot - Pact.Fi - Create Pair



These different tiers are independent of each other and treated as different LPs within Pact.Fi and thus, result in different LP Tokens based on fee tiers. Following point should be considered when enabling support for LP Tokens as collaterals with global borrow limit.

- As global borrow limit is set by the administrator per collateral-borrow asset pair, if the intention of the global borrow limit is to limit the total borrow amount per pair (Example - borrowing of goBTC limited from Pact.Fi LP USDC/ALGO token) the Folks.Finance team should carefully consider the fact that there can be multiple ASAs that represent the same LP combination (USDC/ALGO) just with different fee tiers. In most cases, single LP stands out with the most liquidity and used for majority of the swaps and it would be recommended to support only LP tokens with good liquidity and set global limits with the number of supported LP tokens in mind for a specific pair.

RECOMMENDATION

Review and ensure the global borrow limit which is set through global state *total_borrowed_limit* serves the intended purpose even when there may be multiple LP Tokens available for the same AMM and Liquidity Pool Pairs due to fee tiers.

REGRESSION TESTING COMMENT

13th May 2022 – This issue is closed.

Folks.Finance team acknowledged on the highlighted points and confirmed that there will be separate global borrow limit set for different fee tiers, when LP tokens are being used as collaterals.

VULNERABILITY REFERENCES

Pact.Fi Documentation – Multiple Fee Tiers

https://docs.pact.fi/pact/how-to/pools/multiple-fee-tiers

1.2. Insufficient On-Chain Validation Against Altered Transactions

OBSERVATIONAL ⓘ

VULNERABILITY TRACKING

STATUS: **Closed**

BACKGROUND

As the smart contract code does not validate transaction fields such as group_size of affected transactions, if the front-end web-apps which exist to aid user interaction has been compromised, altered transactions which could be damaging to the users can be forwarded to the user for approval. If not reviewed thoroughly, such transaction groups that are damaging to the user could be approved. Although, the responsibility lies with the user who is approving the transaction group to review each transaction within the atomic group, it is still recommended to have on-chain validations as a safeguard. Do note that this issue only highlights scenarios where users could approve transaction groups which could be damaging to themselves.

DESCRIPTION

Instance 1

Affected File/Code

- oracle_adapter_2_approval_program.py
    - on_add_lp() - Line 168 - 226
    - on_remove_lp() - Line 228-237
    - on_conversion_rate() - Line 244-270

It was noted that above operations were found to lack checks for Global.group_size(). However, given the context of the operations per below, this issue is highlighted as informational only.

- Operations only called by the admin during operations for addition or removal
    - on_add_lp()
    - on_remove_lp()
- Operations that can be called by anyone but needs other transaction groups of Folks.Finance protocol to be used or have any state-changes
    - on_converstion_rate()

Based on the above noted points where these atomic transactions can be grouped together for further integration in the future as a "building block" and the fact that the operations involved are either only approved if sent from a privileged address or the operation only prepares the value for other group transactions as a form of scratch value, serving no purpose if not used with the other transaction groups from Folks.Finance protocol with further validations in place.

RECOMMENDATION

It is generally recommended to clearly define and enforce validations for transaction attributes to check if transaction groups being submitted from the user are according to the specified requirements or documentation. For group size of transactions, if specific transactions are meant to be composable based on the context of the application, further validations could be placed in other transaction within the same transaction groups.

REGRESSION TESTING COMMENT

12th May 2022 – This issue is closed.

Based on discussion with the FolksFinance team, operations were meant to not have group size checks as they are either privileged operations only allowed from the admin's address or a composable transaction which would be part of bigger transaction groups with further validations in place.

VULNERABILITY REFERENCES

Algorand Developer Portal – Guidelines

https://developer.algorand.org/docs/get-details/dapps/avm/teal/guidelines/

Algorand Developer Portal – Transaction References

https://developer.algorand.org/docs/get-details/transactions/transactions/

5. APPENDIX

DISCLAIMER

The material contained in this document is confidential and only for use by the company receiving this information from Vantage Point Security Pte. Ltd. (Vantage Point). The material will be held in the strictest confidence by the recipients and will not be used, in whole or in part, for any purpose other than the purpose for which it is provided without prior written consent by Vantage Point. The recipient assumes responsibility for further distribution of this document. In no event shall Vantage Point be liable to anyone for direct, special, incidental, collateral or consequential damages arising out of the use of this material, to the maximum extent permitted under law.

The security testing team made every effort to cover the systems in the test scope as effectively and completely as possible given the time budget available. There is however no guarantee that all existing vulnerabilities have been discovered due to the nature of manual code review. Furthermore, the security assessment applies to a snapshot of the current state at the examination time.

SCOPE OF AUDIT

Vantage Point reviewed the smart contracts underlying codebase to identify any security or economic flaws, or non-compliance to Algorand best practices. The scope of this review included the following test-cases and audit points.

- Insufficient Sender Address Validation for Privileged Operations
- Lack of Validation for Validity of Referenced States from External Applications
- Insufficient Validation of Transaction Fields and Types
- Validation of RekeyTo address for non-rekeying transactions
- Validation of CloseRemainderTo and AssetCloseTo for non-closing transactions
- Validation of Asset Identifier for Asset Transfer Transactions
- Validation of GroupIndex and GroupSize for Transaction Groups
- Incorrect Order of Operations
- Smart Contract Versions
- Incorrect Use of ScratchVar, Local and Global States
- Flawed/Inaccurate Logical/Mathematical Operations
- Overflow or Underflow Possibilities based on Valid Argument Ranges
- Validation of user-supplied Application Arguments
- Use of Multisignatures for Privileged Accounts
- Other known Algorand Best Practices and Guidelines