SMART CONTRACT REVISION AUDIT PUBLIC REPORT

Revision Audit for Smart Contracts Used in Liquid Governance V2

VPQ-20220292

Folks Finance

2022-06-27

TABLE OF CONTENTS

1. Executive Summary

Overview

Vantage Point Security Pte Ltd was engaged by Folks.Finance to conduct an Algorand smart contract revision audit to changes made to the Folks.Finance Liquid Governance contract as part of the liquid governance upgrade and introduction of the Algo Governance Distributor and Algo Governance Dispenser.

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

- Supporting Documents
  - Supporting documents for audit containing definitions and transaction details
    - Liquid_Governance_V2.pdf
- PyTeal Code
  - Private Repo
    - https://github.com/blockchain-italia/ff-vp-contracts/commits/master
      Commit ID 0363295dd479761f5646621309b2581314721466

Vantage Point performed this review by first understanding the changes to the Folks.Finance protocol in view of changes to Algo Liquidity Governance 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. **Incorrect Validation Logic**
   Based on the composability requirement for Algo Governance Distributor and Algo Governance Dispenser applications, no group size and hardcoded index checks are enforced so that transaction groups can be combined as building blocks for future integration. However, due to incorrectly defined validation logic for close_remainder_to() and rekey_to() against global zero addresses with a specified group transaction index, the smart contract would reject valid transactions with incorrectly enforced validation.

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	**1**	**0**	**1**
Observational	**0**	**0**	**0**
Summary	**1**	**0**	**1**

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
Application Name	Folks Finance Changes to Liquid Governance V2
SVN / GIT Revision Number	0363295dd479761f5646621309b2581314721466
Items Completed	Items Completed • assets/algo_governance/dispenser.json • assets/algo_governance/distributor.json • assets/algo_governance/state.py • algo_governance_dispenser_approval_program.py • algo_governance_distributor_approval_program.py • algo_governance_distributor_clear_program.py • clear_program_6.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
24th June 2022	V0.1	Draft
27th June 2022	V1.0	QA Release

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 for Smart Contracts Used in Liquid Governance V2		**LOW RISK**	⚠
1.1. Incorrect Validation Logic	**Closed**	**LOW RISK**	⚠

4. Detailed Description of Vulnerabilities

2. Revision Audit for Smart Contracts Used in Liquid Governance V2

LOW RISK

1.1. Incorrect Validation Logic

LOW RISK

VULNERABILITY TRACKING

STATUS: **Closed**

BACKGROUND

Smart contracts enforce validations of submitted transactions or transaction groups against the business logic prior to approving transactions which triggers state changes. However, if an incorrect validation logic is implemented in the smart contract, the smart contract may approve or reject transactions that should not be approved or rejected.

DESCRIPTION

Instance 1

Affected File/Code

- algo_governance_distributor_approval_program.py
  - check_user_call() - Line 58-64 is called from
    - on_opt_in()
    - on_close_out()
    - on_mint()
    - on_unmint()
    - on_burn()
    - on_early_claim_rewards()
    - on_claim_rewards()

It was noted that the smart contract does not enforce transaction group size checks and hardcoded transaction index checks for composability by design. However, the implementation of check_user_call() subroutine which incorrectly enforces validation against Gtxn[0].close_remainder_to() and Gtxn[0].rekey_to() while the expected validation given the composability requirement would be Txn.close_remainder_to() and Txn.rekey_to(). Having incorrect validation enforced within the smart contract could cause an availability issue where the smart contract rejects transactions which are legitimate based on the specifications.

Code Snippet - Subroutine check_user_call() - Line 58-65

```
<REDACTED>
 Gtxn[0].close_remainder_to() == Global.zero_address(),
 Gtxn[0].rekey_to() == Global.zero_address()
<REDACTED>
```

RECOMMENDATION

Based on the composability requirement for this smart contract, hard-coded group transaction index such as `Gtxn[0]` could be replaced with `Txn` for the affected part of the code.

REGRESSION TESTING COMMENT

24th June 2022 - This issue is closed.

Based on the commit 3be465eb8387f03e22b553cba9731cdc19109c60, affected lines of code no longer enforces `close_remainder_to()` and `rekey_to()` for the incorrect group transaction index.

VULNERABILITY REFERENCES

Algorand Developer Portal – Transaction Reference

https://developer.algorand.org/docs/get-details/transactions/transactions/?from_query=transaction%20group#common-fields-header-and-type

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