Security Audit Report

Veme Genesis

2/1/2023

PREPARED FOR:
Veme Genesis/Creation Space

ARCADIA CONTACT INFO
Email: audits@arcadiamgroup.com
Telegram: https://t.me/thearcadiagroup

Table of Contents

Executive Summary

A Representative Party of **Veme** ("**CLIENT**") engaged The Arcadia Group ("Arcadia"), a software development, research, and security company, to conduct a review of the following **Veme** smart contracts deployed on the **Polygon address** 0x72E8Bf0DEE1bDbeaC242b4B66b9538AC6F112cB0

The scope of this audit included all contracts (flattened) deployed to the aforementioned address

Arcadia completed this security review using various methods primarily consisting of dynamic and static analysis. This process included a line-by-line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities and limiters, and reference against intended functionality.

There were **2** issues found, of which both were deemed medium severity

Severity Rating	Number of Original Occurrences	Number of Remaining Occurrences
CRITICAL	0	0
HIGH	0	0
MEDIUM	2	2
LOW	0	0
INFORMATIONAL	0	0

Findings

1. The **signVoting** function will use high gas depending on how many users have signed.

Issue: **VemeGenesis**-1
Severity: **Medium**

Target: **VemeGenesis.sol**
Finding Type: **Dynamic**

To validate if a user is already signed, the **signVoting** function is using a loop for all the existing signers.

So the required gas when no signers, 10 signers, 100 signers are all different.

The worst case is if there were lots of signers(ex. over 10k), the transaction could fail due to the gas limit.

Action Recommended:

Use **mapping[votingIndex][user]** to have signed information.

2. The **activateVoting** function will use high gas depending on how many users have signed.

Issue: **VemeGenesis**-1
Severity: **Medium**

Target: **VemeGenesis.sol**
Finding Type: **Dynamic**

The **activeVoting** function has a similar issue with the **signVoting** function.

Action Recommended:

Calculate the cumulative balance when the user signs. This solution is little different from the current activeVoting mechanism, but this kind of algorithm was used in most DAOs.

Conclusion

Arcadia identified issues that occurred on the deployed code, it is expected that redeployment is planned at a later date to correct issues with the voting elements of the contracts.

Disclaimer

While best efforts and precautions have been taken in the preparation of this document, The Arcadia Group and the Authors assume no responsibility for errors, omissions, or damages resulting from the use of the provided information. Additionally, Arcadia would like to emphasize that the use of Arcadia's services does not guarantee the security of a smart contract or set of smart contracts and does not guarantee against attacks. One audit on its own is not enough for a project to be considered secure; that categorization can only be earned through extensive peer review and battle testing over an extended period.