grapevine
WORLD

SMART CONTRACTS AUDIT

by

VENTURE BOOST

SUMMARY

This document summarizes the assessment and conclusions regarding the overall quality, security and correctness of the Grapevine World smart contracts as evaluated by VentureBoost. The scope of this audit was a second analysis and documentation of the updated Grapevine World smart contracts codebase for quality, security and correctness.

The VentureBoost smart contract team have performed a second audit of the Grapevine smart contracts. Our findings are detailed below.

VentureBoost have no stake or vested interest in Grapevine and have made best efforts to provide an objective analysis.

This audit does not represent an endorsement of the reliability or effectiveness of the smart contracts but constitutes merely an assessment of its logic and implementation. In order to ensure that a secure contract is able to withstand the fast-paced and rapidly changing environment of the Ethereum network, we recommend to put in place a bug bounty program to encourage further and active analysis of the smart contract.

The statements made in this document should not be interpreted as investment or legal advise and its authors shall not be held liable for any decisions made based on them.

AUDIT GOALS AND FOCUS

Sound Architecture

This audit includes assessments of the overall architecture and design choices. Given the subjective nature of these assessments, it will be up to the Grapevine development team to address any of the issues that have been outlined and to make any changes to the smart contracts.

Smart Contract Best Practices

This audit evaluates whether the codebase follows the current established best practices for smart contract development as defined by OpenZeppelin and others.

Code Correctness

This audit evaluates whether the code performs and executes what it is intended to do.

Code Quality

This audit evaluates whether the code has been written in a way that ensures readability and maintainability.

Security

This audit aims to identity any exploitable security vulnerabilities or other potential threats to either the operator of the token sale or its participants.

Testing and Testability

This audit examines how easily the code can be tested and reviews how thoroughly tested the code is.

INTRODUCTION

Source Code

The Grapevine Token and token sale source code was made available in a git (GitHub) repository:

https://github.com/GrapevineWorld/crowdsale-contracts/blob/master/flatContracts/FlatGrapevineWhitelist.sol

The following commit of the git repository was audited:

https://github.com/GrapevineWorld/crowdsale-contracts/commit/aee094ef913553dadd56da03b2c53fcfce34de00

We have not audited the library code as a whole, any code for the accounts that will be recipients of the funds of the crowdsale or any code involved in subsequent allocation of tokens to users.

Contracts

The following contracts have been assessed as part of this audit:

- **FlatGrapevineWhitelist.sol**

Testing

Automated unit tests were provided for all contracts with 100% coverage.

ABOUT THE GRAPEVINE WHITELISTING SMART CONTRACT

1. The Grapevine whitelisting contract supports both off-chain and on-chain whitelisting.

2. Participants can be whitelisted off-chain with the help of elliptic curve signatures.

3. The ICO owner can also whitelist participants manually by adding their address to the whitelist.

General Notes

- The whitelisting contract is structured in a manner that ICO owner can whitelist addresses both off-chain and on-chain. Off-chain whitelisting requires only one transaction while on-chain whitelisting requires two transactions. The off-chain whitelisting method is preferred as long as participants do not suffer from the inconvenience of sending signed data along with the contribution transaction.

TERMINOLOGY

This audit uses the following terminology:

Likelihood

How likely a bug is to be encountered or exploited in the wild, as specified by the OWASP Risk Rating Methodology:

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology#Step_2:_Factors_for_Estimating_Likelihood

Impact

The impact a bug would have if exploited, as specified by the OWASP Risk Rating Methodology:

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology#Step_3:_Factors_for_Estimating_Impact

Severity

How serious an issue is, derived from Likelihood and Impact, as specified by the OWASP Risk Rating Methodology:

https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology#Step_4:_Determining_the_Severity_of_the_Risk

AUDIT RESULTS AND FINDINGS

Note Issues

None found.

Low Severity Issues

None found.

Medium Severity Issues

None found.

High Severity Issues

None found.

Critical Issues

None found.

CONTRACT ISSUE RATING

CONCLUSIONS AND RECOMMENDATIONS

The Grapevine smart contracts code is heavily based on the OpenZeppelin open source library which is best practice. We did not find any note, low, medium, high severity or critical issues. While whitelisting is generally neither advisable nor good practice, overall, the code is sophisticated and well commented.