GetSmart

WDXStacking Smart Contract Audit

If you have any questions about smart-contract audit, contact us hello@getsmart.site

17.09.2020 www.getsmart.site

TABLE OF CONTENTS

INTRODUCTION

Our company provides comprehensive, independent smart contract auditing. We help stakeholders confirm the quality and security of their smart contracts using our standardized audit process. The scope of this audit was to analyze and document the WDXStacking contract.

AUDIT METHODOLOGY

WDXStacking contracts audit consist of four categories of analysis.

1. Design Patterns

We inspect the structure of the smart contract, including both manual and automated analysis.

Contract that was written by developers is not well commented (in the most of functions comments are missed).

Code is not unified (some times is used shortcut “uint” for the “uint256” type, sometimes is used “uint256” type name itself.

2. Static Analysis

The static analysis is performed using a series of automated tools, purposefully designed to test the security of the contract.

All the issues found by tools were manually checked (rejected or confirmed).

3. Security Issues Analysis

Contract reviewing to identify common vulnerabilities.

- Re-Entrancy, a critical flaw where one contract exploits the execution state of another contract. Overall Severity: Critical

Not found.

- Self-Destructing Contract, a flaw in how a library contract delegates its functions to smart contracts that invoke it. Overall Severity: Critical

Not found.

- Transaction-Ordering Dependence, an uncommon flaw that allows a miner to manipulate a transaction's output by its timestamp. Overall Severity: Low

Not found.

- Timestamp Dependency, a bug that changes the result of a transaction depending on when it executes within a block. Overall Severity: Medium

Not found.

Contract time management cannot be affected in a critical way by possible 10-15 minutes timestamp manipulations.

- Assertion Failure, an indication that another, potentially critical flaw occurred upstream. Overall Severity: Medium

Not found.

4. Manual Analysis

Comparing of requirements and implementation. Reviewing of a smart contract for compliance with specified customer requirements. Checking for a gas optimization and self-documentation. Running tests of the properties of the smart contract in test net.

Some functions miss user allowance and token balance validation (line 168, function sendToStaking has allowance validation but has not token balance validation, line 146 function buyStatus has not both allowance and token balance validation). If contract of

used token follows the ERC20 standard it is not critical but is desirable for the better user experience.

6. Contracts Reviewed

On September 17, 2020 following smart contract was reviewed:

Contract name: WDXStacking
Compiler version 0.5.17, overall 249 lines of code.

AUDIT SUMMARY

The contracts have been found to be free of security issues.

Contract has well-formed structure. Gas usage is optimal.

Adding user input validation is recommended.

Category of analysis	Result
Design Patterns	Updates Recommended
Security Issues Analysis	Passed
Static Analysis	Passed
Manual Analysis	Updates Recommended

Test Results.

All tests run successfully with excellent code coverage.

ISSUES DISCOVERED

Issues are listed from most critical to least critical. Severity is determined by an assessment of the risk of exploitation or otherwise unsafe behavior.

Severity Levels

- Informational - No impact on the contract.
- Low - Minimal impact on operational ability.
- Medium - Affects the ability of the contract to operate.
- High - Affects the ability of the contract to work as designed in a significant way.
- Critical - Funds may be allocated incorrectly, lost or otherwise result in a significant loss.

Issues.

1. Informational: There are not enough comments in the code.
   Recommended: Add comments to the contract functions.
2. Informational: Code is not unified.
   Recommended: Use either shortcuts “uint” or full type name “uint256”.
3. Informational: Missed user input validation.
   Recommended: Add user balance and allowance validation to the functions line 146 function buyStatus , line 168 function sendToStaking.

CONCLUSION

The WDXStacking smart contracts are well crafted, following common security practices. Full set of tests were provided to enforce correctness.