COINMERCENARY / SMART CONTRACT AUDIT

Bayan Token

24 OCTOBER 2018 / TABLE OF CONTENTS

INTRODUCTION

CoinMercenary provides comprehensive, independent smart contract auditing.

We help stakeholders confirm the quality and security of their smart contracts using our comprehensive and standardized audit process. Each audit is unbiased and verified by multiple reputable auditors.

The scope of this audit was to analyze and document the Bayan token contract.

This audit provides practical assurance of the logic and implementation of the contract.

AUDIT METHODOLOGY

CoinMercenary audits consist of four categories of analysis.

Design Patterns

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

The design pattern analysis checks appropriate test coverage, utilizes a linter to ensure consistent style and composition, and code comments are reviewed. Overall architecture and safe usage of third party smart contracts are checked to ensure the contract is structured in a way that will not result in future issues.

Static Analysis

The static analysis portion of our audit is performed using a series of automated tools, purposefully designed to test the security of the contract. These tools include:

- **Manticore** - Dynamic binary analysis tool with EVM support.
- **Mythril** - Reversing and bug hunting framework for the Ethereum blockchain.
- **Oyente** - Analyzes Solidity code to find common vulnerabilities.
- **Solgraph** - DOT graph creation for visualizing function control flow of a Solidity contract to highlight potential security vulnerabilities.

Data flow and control flow are also analyzed to identify vulnerabilities.

Manual Analysis

Performing a hands on review of the smart contract to identify common vulnerabilities is the most intensive portion of our audit. Checks for race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks are part of our standardized process.

Network Behavior

In addition to our design pattern check, we also specifically look at network behavior. We model how the smart contract will operate once in production,

then determine the answers to questions such as: how much gas will be used, are there any optimizations, how will the contract interact?

Contracts Reviewed

On October 24th, 2018 the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
BayanToken.sol	f601454a3a7c21260ecf32b45e212c002e7685ae9ef1cafbb538b7e2928ad201

Remediation Audit #1

On November 5th, 2018 the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
BayanToken.sol	b8dc6089f00f03ebe65cf726d155b939e2968225b54bec3ade49c21bf1518a80

AUDIT SUMMARY

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

Analysis Results

	Initial Audit	Remediation Audit
Design Patterns	Passed	Passed
Static Analysis	Updates Recommended	Passed
Manual Analysis	Passed	Passed
Token Allocation	Updates Recommended	Passed
Network Behavior	Updates Recommended	Passed

Test Results

- No unit test coverage available.

Token Allocation Results

- Symbol: BYT
- Decimal: 2
- Total Supply: 200,000,000

Explicit Vulnerability Check Results

Known Vulnerability	Results
Parity Multisig Bug 2	Not vulnerable
Callstack Depth Attack	Not vulnerable
Transaction-Ordering Dependence	Not vulnerable
Timestamp Dependency	Not vulnerable
Re-Entrancy Vulnerability	Not vulnerable
Proxy and Buffer Overflow	Not vulnerable

ISSUES DISCOVERED

Issues below 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

BT-1 / Low: Transfer event is not emitted during initial assignment of token balance

Present in BayanToken.sol, line 1477

Explanation

Several third party services such as Etherscan use the Transfer event to maintain an external record of transfers. When assigning the initial tokens in the constructor of the token contract, a Transfer event must be emitted to ensure proper tracking of token allocation when viewed using these third party services.

Example:

```
    emit Transfer(address(0), msg.sender, initialSupply);
```

Resolution

Resolved in remediation audit #1.

BT-2 / Low: Use latest Solidity compiler version

Present in all contract files

Explanation

Update all contract files to use the latest version of Solidity compiler in order to ensure the latest performance enhancements, features and bug fixes are available.

Resolution

Resolved in remediation audit #1.

BT-3 / Informational: Defining constructors as functions with the same name as the contract is deprecated

Present in BayanToken.sol, lines 1476

Explanation

Defining constructors as functions with the same name as the contract is deprecated. Use "constructor(...) { ... }" instead.

Resolution

Resolved in remediation audit #1.

BT-4 / Informational: Invoking events without “emit” prefix is deprecated

Present in BayanToken.sol, lines 1519, 1517, 1490, 1436, 1414, 1387, 1370, and 1328

Explanation

All events must be prefixed with “emit” prior to being invoked. Example:

```
    emit Transfer(address(0), msg.sender, initialSupply);
```

Resolution

Resolved in remediation audit #1.

CONCLUSION

October 29th, 2018

CoinMercenary provides comprehensive, independent smart contract auditing.

We help stakeholders confirm the quality and security of their smart contracts using our comprehensive and standardized audit process. Each audit is unbiased and verified by multiple reputable auditors.

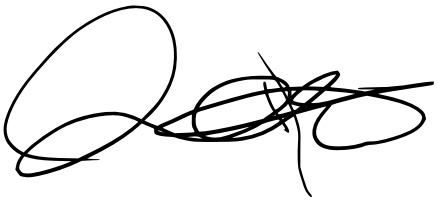
The scope of this audit was to analyze and document the Bayan token contract. The audit provides practical assurance of the logic and implementation of the contracts.

CoinMercenary has reviewed the Bayan smart contract and found it to be free of security issues and logic errors.

The audit began on October 23rd, ending on November 5th. Two “low” level issues and two “informational” level issues were documented.

Working with the Bayan team has been a pleasure and we look forward to seeing their continued success.

Sincerely,



JONATHAN GEORGE