CERTIK

Security Assessment

BoringDAO

May 29th, 2021

Table of Contents

Summary

This report has been prepared for BoringDAO smart contracts, to discover issues and vulnerabilities in the source code of their Smart Contract as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Addtionally, this audit is based on a premise that all external contracts were implemented safely.

The security assessment resulted in 6 findings that ranged from minor to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases given they are currently missing in the repository;
- Provide more comments per each function for readability, especially contracts are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	BoringDAO
Platform	Ethereum, BSC
Language	Solidity
Codebase	https://github.com/BoringDAO/boringDAO-contract/blob/master/contracts/token/Boring.sol
Commits	<e2f3ba38f37faa9afa9722b56995fae5ffade58a>

Audit Summary

Delivery Date	May 29, 2021
Audit Methodology	Manual Review
Key Components	

Vulnerability Summary

Total Issues	6
● Critical	0
● Major	0
● Medium	0
● Minor	1
● Informational	5
● Discussion	0

Audit Scope

ID	file	SHA256 Checksum
BDA	Boring.sol	1a64eec53247f97b97ae0100692209e7f95bbd3bc98cf247b68c50f0fc02b7fb

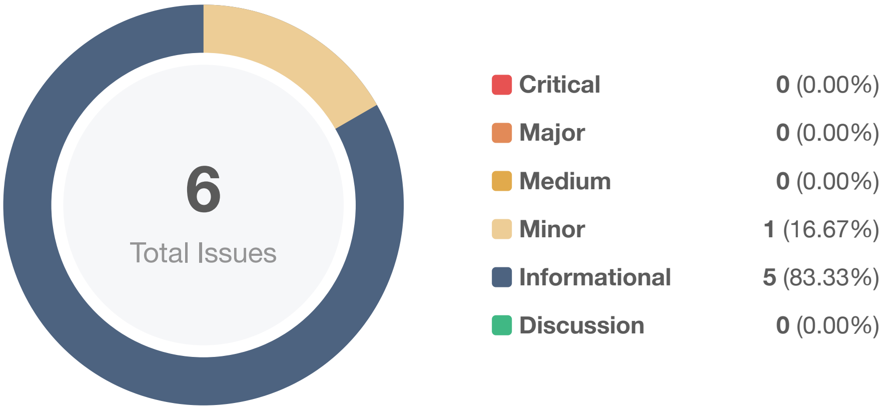
Centralization Roles

The BoringDAO smart contract introduces an authorization.

Owner:

setSwitchOn(): Set the value of `switchOn`. Only when `switchOn == true`, uses can call the `toBar()` function to exchange `BOR` tokens.

Findings



ID	Title	Category	Severity	Status
BDA-01	Proper Usage of "public" And "external" Type	Gas Optimization	● Informational	✓ Resolved
BDA-02	Missing Emit Events	Coding Style	● Informational	✓ Resolved
BDA-03	Missing Zero Address Validation	Logical Issue	● Informational	✓ Resolved
BDA-04	Verify Before Operation	Gas Optimization	● Informational	✓ Resolved
BDA-05	Boolean Equality	Coding Style	● Informational	✓ Resolved
BDA-06	Privileged Ownership	**Centralization / Privilege**	● **Minor**	ⓘ **Acknowledged**

BDA-01 | Proper Usage of "public" And "external" Type

Category	Severity	Location	Status
Gas Optimization	● Informational	Boring.sol: 27, 32, 39	✓ Resolved

Description

`public` functions that are never called by the contract could be declared `external`.

Recommendation

Consider using the `external` attribute for functions never called from the contract.

Alleviation

The team heeded our advice and changed related code. Code change was applied in commit 50ce93a506039e54a71d25b32a7d6942ecedb432.

BDA-02 | Missing Emit Events

Category	Severity	Location	Status
Coding Style	● Informational	Boring.sol: 27	✓ Resolved

Description

Some functions should be able to emit event as notifications to customers because they change the status of sensitive variables.

Recommendation

Consider adding an emit after changing the status of variables.

Alleviation

The team heeded our advice and changed related code. Code change was applied in commit 50ce93a506039e54a71d25b32a7d6942ecedb432.

BDA-03 | Missing Zero Address Validation

Category	Severity	Location	Status
Logical Issue	● Informational	Boring.sol: 19	✓ Resolved

Description

Addresses should be checked before assignment to make sure they are not zero addresses.

Recommendation

Consider adding a check like below:

constructor():

```
require(_bor != address(0), "_bor address cannot be 0");
```

Alleviation

The team heeded our advice and changed related code. Code change was applied in commit 50ce93a506039e54a71d25b32a7d6942ecedb432.

BDA-04 | Verify Before Operation

Category	Severity	Location	Status
Gas Optimization	● Informational	Boring.sol: 50	✓ Resolved

Description

We recommend verifying the input parameters before any operation, not only to avoid input errors, but also to save gas.

Recommendation

Consider modifying the function like below:

```
function _beforeTokenTransfer(address from, address to, uint256 amount) internal override {
        require(to != address(this), "ERC20: transfer to the token contract address");
        super._beforeTokenTransfer(from, to, amount);
    }
```

Alleviation

The team heeded our advice and changed related code. Code change was applied in commit 50ce93a506039e54a71d25b32a7d6942ecedb432.

BDA-05 | Boolean Equality

Category	Severity	Location	Status
Coding Style	● Informational	Boring.sol: 54	✓ Resolved

Description

`switchOn` is a `bool` type state variable, which can be directly used as an expression result in require.

Recommendation

Consider modifying the `onlySwitchOn` modifier like below:

```
modifier onlySwitchOn {
        require(switchOn, "only switchOn true");
        _;
    }
```

Alleviation

The team heeded our advice and changed related code. Code change was applied in commit 50ce93a506039e54a71d25b32a7d6942ecedb432.

BDA-06 | Privileged Ownership

Category	Severity	Location	Status
Centralization / Privilege	● Minor	Boring.sol: 39	ⓘ Acknowledged

Description

The owner of contract `Boring` has the permission to:

1. set the value of `switchOn`, only when `switchOn==true`, uses can call the `toBar()` function to exchange `BOR` tokens;

without obtaining the consensus of the community.

Recommendation

Renounce ownership when it is the right timing, or gradually migrate to a timelock plus multisig governing procedure and let the community monitor in respect of transparency considerations.

Alleviation

Customer team response:
The `owner` will be transferred to a timelock contract in the future.

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

CERTIK