CERTIK

Security Assessment

Tipsy Online

Jun 11th, 2021

Table of Contents

Summary

This report has been prepared for Tipsy Online 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 Static Analysis and 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.

The security assessment resulted in findings that ranged from critical 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	Tipsy Online
Platform	BSC
Language	Solidity
Codebase	https://bscscan.com/address/0x68c4d87bbf94379d6c94466cf3ad70d53603fd1e#code
Commit	

Audit Summary

Delivery Date	Jun 11, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

Total Issues	3
● Critical	0
● Major	1
● Medium	0
● Minor	1
● Informational	1
● Discussion	0

Audit Scope

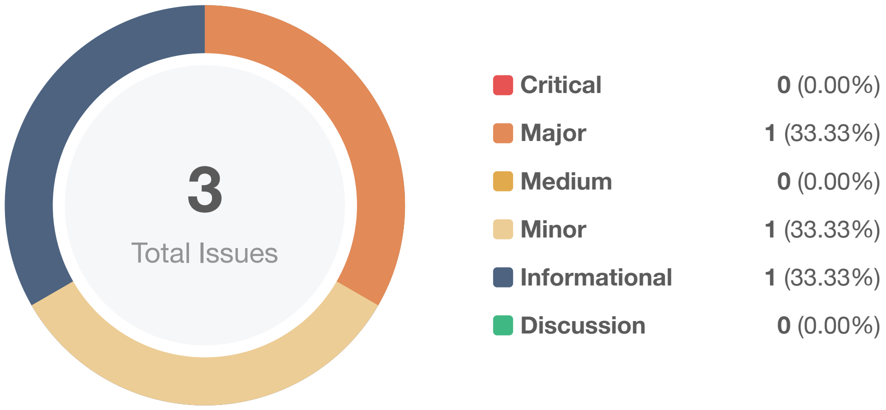
ID	file	SHA256 Checksum
CTC	CoinToken.sol	66aeae1850e8cf7586871f889371f7c4a252d684dbeead7158534f5f6f092125

To set up the project correctly, improve overall project quality and preserve upgradability, the role `owner` is adopted to call following functions in the codebase:

- `owner` is adopted to call `transferOwnership()` in contract `Ownable`;
- `owner` is adopted to call `pause()` and `unpause()` in contract `Pausable`;
- `owner` is adopted to call `blackListAddress()` in contract `PausableToken`;
- `owner` is adopted to call `mint()` in contract `CoinToken`;

To improve the trustworthiness of the project, dynamic runtime updates in the project should be notified to the community. Any plan to invoke the aforementioned functions should also be considered to move to the `Timelock` contract execution queue.

Findings



ID	Title	Category	Severity	Status
CTC-01	Lack of Error Message	Logical Issue	● Informational	ⓘ Pending
CTC-02	Centralization Risks	Logical Issue	● Major	ⓘ Acknowledged
CTC-03	Incorrect Checking Function	Logical Issue	● Minor	ⓘ Pending

CTC-01 | Lack of Error Message

Category	Severity	Location	Status
Logical Issue	● Informational	CoinToken.sol: 46, 56, 74, 82, 130~132, 147~150, 192, 253	ⓘ Pending

Description

The error message in `require` checking can indicate the desired operation failure to users or relay essential warnings:

- `require(msg.sender == owner);`
- `require(newOwner != address(0));`
- `require(!paused);`
- `require(paused);`
- `require(tokenBlacklist[msg.sender] == false);`
- `require(_to != address(0));`
- `require(_value <= balances[msg.sender]);`
- `require(tokenBlacklist[msg.sender] == false);`
- `require(_to != address(0));`
- `require(_value <= balances[_from]);`
- `require(_value <= allowed[_from][msg.sender]);`
- `require(tokenBlacklist[_address] != _isBlackListed);`
- `require(_value <= balances[_who]);`

Recommendation

We advise the client to provide an error message string for the `require` checking.

CTC-02 | Centralization Risks

Category	Severity	Location	Status
Logical Issue	● Major	CoinToken.sol: 260	ⓘ Acknowledged

Description

The role `owner` has the authority to `mint()` any additional tokens after `CoinToken` deployment.

Recommendation

We advise the client to handle the `owner` carefully to avoid any potential hack. We also advise the client to consider the following solutions:

1. `Timelock` with reasonable latency for community awareness on privileged operations;
2. Multisig with community-voted 3rd-party independent co-signers;
3. DAO or Governance module increasing transparency and community involvement;

Alleviation

[Tipsy]: The contract is made with mint functionality because of the following reasons:

- burn unsold tokens
- able to mint the token back if any mistakes.

The team is planned to burn the tokens after the private sale is completed by early July 2021. Also, renounce the ownership and send it to address zero.

CTC-03 | Incorrect Checking Function

Category	Severity	Location	Status
Logical Issue	● Minor	CoinToken.sol: 13, 25, 31	ⓘ Pending

Description

The `assert` function should only be used to test for internal errors and to check invariants according to the docs. For example,

- `assert(c / a == b);`
- `assert(b <= a);`
- `assert(c >= a);`

Recommendation

We recommend using `require` instead of `assert` at the aforementioned lines.

Appendix

Finding Categories

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.

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.