# Candy Token

## Security Assessment

December 12th, 2020

For :
Candy Token

By :
Alex Papageorgiou @ CertiK
alex.papageorgiou@certik.org

Georgios Delkos @ CertiK
georgios.delkos@certik.io

# Disclaimer

CertiK reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are 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 review.

CertiK Reports do 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.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing 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.

## What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has indeed completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.

# Overview

## Project Summary

| Project Name | Candy Token |
| --- | --- |
| **Description** | A typical ERC20 implementation with enhanced features. |
| **Platform** | Ethereum; Solidity, Yul |
| **Codebase** | N/A |
| **Commits** | N/A |

## Audit Summary

| Delivery Date | December 12th, 2020 |
| --- | --- |
| **Method of Audit** | Static Analysis, Manual Review |
| **Consultants Engaged** | 2 |
| **Timeline** | December 12th, 2020 - December 12th, 2020 |

## Vulnerability Summary

| Total Issues | 17 |
| --- | --- |
| **Total Critical** | 0 |
| **Total Major** | 0 |
| **Total Medium** | 1 |
| **Total Minor** | 3 |
| **Total Informational** | 13 |

# Executive Summary

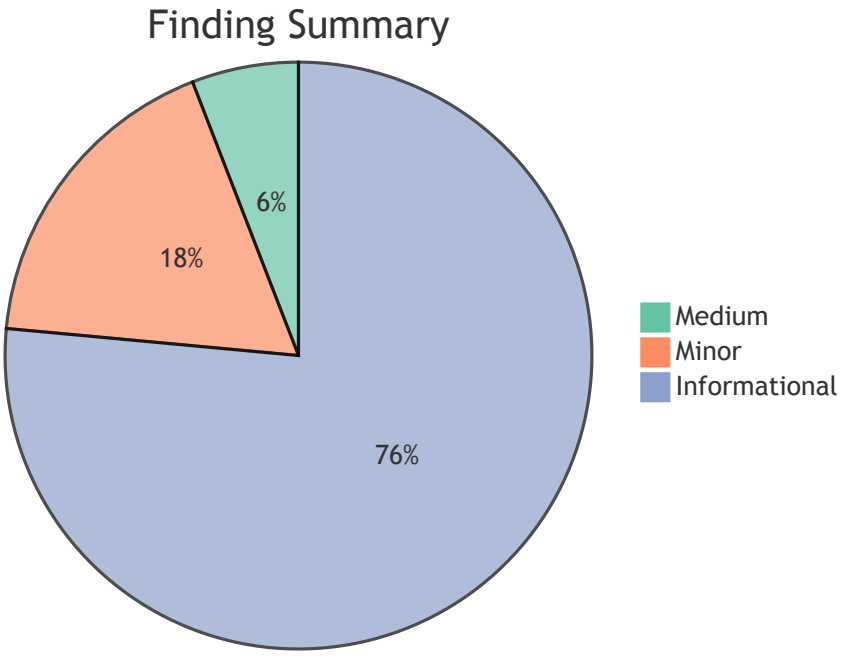
During the course of the audit, we identified flaws in the code structure as well as other best practices issues along with a complete centralization factor in the way tokens can be burned or minted from an arbitrary address by the owner. We relayed them to the CAD team who stated that they will not alleviate any of the issues found during the audit as they deem them non-critical to the overall security of the project and that the centralization aspect is desired and a trait of the protocol itself.

# Files In Scope

| ID | Contract | Location |
| --- | --- | --- |
| CAD | cad.sol | cad.sol |


# File Dependency Graph (BETA)

# Findings




| ID | Title | Type | Severity | Resolved |
|---|---|---|---|---|
| CAD-01 | Unlocked Compiler Version | Language Specific | Informational | |
| CAD-02 | Compare To Constant Boolean | Language Specific | Informational | |
| CAD-03 | Functions That Could Be Declared External | Language Specific | Informational | |
| CAD-04 | Redundant Fallback Function | Logical Issue | Minor | |
| CAD-05 | Visibility Specifiers Missing | Language Specific | Informational | |
| CAD-06 | Require Missing | Language Specific | Informational | |

| | | | | |
|---|---|---|---|---|
| | Error Message | | |  |
| CAD-07 | Redundant Check | Logical Issue | Informational |  |
| CAD-08 | Owner Has Complete Control. | Centralization | Informational |  |
| CAD-09 | Missing Check | Logical Issue | Informational |  |
| CAD-10 | Assert Over Require | Gas Optimization | Minor |  |
| CAD-11 | Magic Numbers | Coding Style | Informational |  |
| CAD-12 | Input Sanitazation Missing | Logical Issue | Medium |  |
| CAD-13 | Restricted Functionality | Logical Issue | Minor |  |
| CAD-14 | Unused Variables | Language Specific | Informational |  |
| CAD-15 | Ambiguous Comment | Coding Style | Informational |  |
| CAD-16 | Redundant Statements | Dead Code | Informational |  |
| CAD-17 | Public Over Internal | Language Specific | Informational |  |

# CAD-01: Unlocked Compiler Version

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | cad.sol L5 |

## Description:

The contract has unlocked compiler version. An unlocked compiler version in the source code of the contract permits the user to compile it at or above a particular version. This, in turn, leads to differences in the generated bytecode between compilations due to differing compiler version numbers. This can lead to an ambiguity when debugging as compiler specific bugs may occur in the codebase that would be hard to identify over a span of multiple compiler versions rather than a specific one.

## Recommendation:

We advise that the compiler version is instead locked at the lowest version possible that the contract can be compiled at. For example, for version `v0.6.2` the contract should contain the following line:

```
pragma solidity 0.6.2;
```

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-02: Compare To Constant Boolean

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | cad.sol L120, L121, L124, L154, L174 |

## Description:

The code compares to a constant boolean value.

## Recommendation:

Refactor the code and remove the constant boolean value.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-03: Functions That Could Be Declared External

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | cad.sol L133, L138, L142, L146, L166, L303, L350, L362 |

## Description:

Code contains functions that could be declared as external.

## Recommendation:

Refactor the code and make the functions external.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-04: Redundant Fallback Function

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Minor | cad.sol L340 |

## Description:

The code contains a redundant fallback function.

## Recommendation:

Refactor the code and remove the current fallback function.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-05: Visibility Specifiers Missing

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | cad.sol L14, L214 |

## Description:

The linked variable declarations do not have a visibility specifier explicitly set.

## Recommendation:

Inconsistencies in the default visibility the Solidity compilers impose can cause issues in the functionality of the codebase. We advise that visibility specifiers for the linked variables are explicitly set.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-06: Require Missing Error Message

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | cad.sol L31, L40, L45, L121, L124, L232, L233, L261-L263, L282, L304, L331, L336, L341 |

## Description:

The linked statements do not affect the functionality of the codebase and appear to be either leftovers from test code or older functionality.

## Recommendation:

We advise that they are removed to better prepare the code for production environments.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-07: Redundant Check

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Informational | cad.sol L233, L262, L263, L304 |

## Description:

Code contains checks that are redundant.

## Recommendation:

Refactor the code and remove the redundant checks as the code uses SafeMath.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-08: Owner Has Complete Control.

| Type | Severity | Location |
| --- | --- | --- |
| Centralization | Informational | cad.sol L303 |

## Description:

Contract owner has complete control of every functionality in the contract burn/mint etc.

## Recommendation:

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-09: Missing Check

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Informational | cad.sol L79 |

## Description:

The code has commented out a check.

## Recommendation:

Refactor the code and add (require != 0, Error message).

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-10: Assert Over Require

| Type | Severity | Location |
| --- | --- | --- |
| Gas Optimization | Minor | cad.sol L71, L89, L98 |

## Description:

The code uses asser over require.

## Recommendation:

Refactor the code and use `require` over `assert` statements, as a failed `assert` statement consumes all the remaining gas.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-11: Magic Numbers

| Type | Severity | Location |
| --- | --- | --- |
| Coding Style | Informational | cad.sol L219, L258, L275 |

## Description:

The contains literals with no description.

## Recommendation:

Refactor the code and add descriptions.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-12: Input Sanitazation Missing

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Medium | cad.sol L257-L272, L274-L287, L303-L309, L350-L356 |

## Description:

Inexistent input sanitization, as parameters of type `address` should not be equal to the zero address.

## Recommendation:

Refactor the code and add checks.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-13: Restricted Functionality

| Type | Severity | Location |
| --- | --- | --- |
| Logical Issue | Minor | cad.sol L274-L287 |

## Description:

Restricting functionlity, as the user cannot increase/decrease a non zero allownce.

## Recommendation:

Refactor the code and consider adding such functionality.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-14: Unused Variables

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | cad.sol L371-L373 |

## Description:

Unused Variables

## Recommendation:

Refactor the code and remove all unused variables.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-15: Ambiguous Comment

| Type | Severity | Location |
| --- | --- | --- |
| Coding Style | Informational | cad.sol L375 |

## Description:

Ambiguous Comment

## Recommendation:

Refactor the code.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-16: Redundant Statements

| Type | Severity | Location |
| --- | --- | --- |
| Dead Code | Informational | cad.sol L378 |

## Description:

The linked statements do not affect the functionality of the codebase and appear to be either leftovers from test code or older functionality.

## Recommendation:

We advise that they are removed to better prepare the code for production environments.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# CAD-17: Public Over Internal

| Type | Severity | Location |
| --- | --- | --- |
| Language Specific | Informational | cad.sol L111-L112 |

## Description:

The code declares public visibility for mappings that have getter functions.

## Recommendation:

Refactor the code and use internal as the mappings have getter functions L138-L144.

## Alleviation:

The Candy Token development team has acknowledged this exhibit but decided to not apply its remediation in the current version of the codebase due to time constraints.

# Appendix

## Finding Categories

### Gas Optimization

Gas Optimization findings refer to exhibits that 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.

### Mathematical Operations

Mathematical Operation exhibits entail findings that relate to mishandling of math formulas, such as overflows, incorrect operations etc.

### Logical Issue

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

### Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

### Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

### Data Flow

Data Flow findings describe faults in the way data is handled at rest and in memory, such as the result of a `struct` assignment operation affecting an in-memory `struct` rather than an in-storage one.

### Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of `private` or `delete`.

### Coding Style

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

### Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code, such as a `constructor` assignment imposing different `require` statements on the input variables than a setter function.

### Magic Numbers

Magic Number findings refer to numeric literals that are expressed in the codebase in their raw format and should otherwise be specified as `constant` contract variables aiding in their legibility and maintainability.

### Compiler Error

Compiler Error findings refer to an error in the structure of the code that renders it impossible to compile using the specified version of the project.

### Dead Code

Code that otherwise does not affect the functionality of the codebase and can be safely omitted.