Audit Report

for

Bunzz ERC4671

Author: Kazune Takeda

Mar 8th, 2023

Table of Contents

Summary

Vulnerability Severity Classification

Overview

Project Summary

Audit Summary

Vulnerability Summary

Findings

NTT-01 : Local Variable Shadowing

NTT-02 : Unsafe Owner

NTT-03 : Unused Functions

NTT-04 : Optimizable Logical Operations

NTT-05 : Unusual Contract Name

Appendix

Summary

Non-Tradable Token (ERC4671) represents a module that a project may need when is looking to create tokens that are non-tradable, basically, tokens that cannot be transferred from one wallet to another,

This module has the feature of minting as many tokens as you want and hosting their metadata on a centralized api, in addition, the tokens cannot be transferred to any address, making them non-tradable tokens.

Vulnerability Severity Classification

To standardize the evaluation, we define the following terminology based on OWASP Risk Rating Methodology:

- **Likelihood** represents how likely a particular vulnerability is to be uncovered and exploited in the wild;
- **Impact** measures the technical loss and business damage of a successful attack;
- **Severity** demonstrates the overall criticality of the risk.

Likelihood and **impact** are categorized into three ratings: H, M and L, i.e., high, medium and low respectively. Severity is determined by likelihood and impact and can be classified into four categories accordingly, i.e., Critical, High, Medium, Low shown in table below:

Impact	*High*	Critical	Major	Medium
	Medium	Major	Medium	Minor
	Low	Medium	Minor	Informational
		High	*Medium*	*Low*
			Likehood	

Overview

Project Summary

Project Name	ERC4671
Platform	Bunzz
Language	Solidity
Codebase	https://app.bunzz.dev/module-templates/5c7bc93c-b2b7-4d11-9be7-fd62fa9a22b1/code

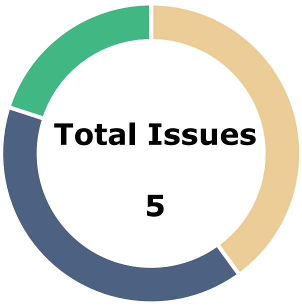
Audit Summary

Delivery Date	Mar 8, 2023
Audit Methodology	Static Analysis, Manual Review
Key Components	TokenERC4671, IERC4671, IERC4671Enumerable, IERC4671Metadata

Vulnerability Summary

Total Issues	5
● Critical	0
● Major	0
Medium	0
● Minor	2
● Informational	2
● Discussion	1

Findings



Critical	**0** (0.00%)
Major	**0** (0.00%)
Medium	**0** (0.00%)
Minor	**2** (40.00%)
Informational	**2** (40.00%)
Discussion	**1** (20.00%)

ID	Title	Category	Severity	Status
NTT-01	Local Variable Shadowing	Volatile Code	● Minor	Pending
NTT-02	Unsafe Owner	Control Flow	● Informational	Pending
NTT-03	Unused Functions	Coding Style	● Informational	Pending
NTT-04	Optimizable Logical Operations	Gas Optimization	● Informational	Pending
NTT-05	Unusual Contract Name	Coding Style	● Discussion	Pending

NTT-01 | Local Variable Shadowing

Category	Severity	Location	Status
Volatile Code	● Minor	https://app.bunzz.dev/module-templates/5c7bc93c-b2b7-4d11-9be7-fd62fa9a22b1/code/TokenERC4671.sol#L81	ⓘ Pending

Description

TokenERC4671.tokenURI(uint256).baseURI (TokenERC4671.sol#L81) shadows:

- TokenERC4671.baseURI (TokenERC4671.sol#L30) (state variable)

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Recommendation

Rename the local variables that shadow another component.

Alleviation

NTT-02 | Unsafe Owner

Category	Severity	Location	Status
Control Flow	● Minor	https://app.bunzz.dev/module-templates/5c7bc93c-b2b7-4d11-9be7-fd62fa9a22b1/code/TokenERC4671.sol#L38	ⓘ Pending

Description

TokenERC4671 contract is using onlyCreator role for mint and revoke functions.

The creator in this contract is usually the deployer of this contract, but when you lose access to the owner wallet address, you will no longer control this contract.

This could lead you no longer able to control this contract.

Recommendation

Use Openzeppelin's Ownable contract instead. Ownable2Step introduced from Openzeppelin v4.8 would be more secure.

Alleviation

NTT-03 | Unused Functions

Category	Severity	Location	Status
Dead Code	● Informational	https://app.bunzz.dev/module-templates/5c7bc93c-b2b7-4d11-9be7-fd62fa9a22b1/code/TokenERC4671.sol#L168-L187	ⓘ Pending

Description

TokenERC4671._removeFromUnorderedArray(uint256[],uint256) (TokenERC4671.sol#L168-L172) is never used and should be removed

TokenERC4671._removeToken(uint256) (Flatten.sol#L175-L187) is never used and should be removed

Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Recommendation

Rename unused functions.

Alleviation

NTT-04 | Optimizable Logical Operations

Category	Severity	Location	Status
Gas Optimization	● Informational	https://app.bunzz.dev/module-templates/5c7bc93c-b2b7-4d11-9be7-fd62fa9a22b1/code/TokenERC4671.sol#L134, #L150, #L163, #L180, #L184	ⓘ Pending

Description

#L134: _numberOfValidTokens[token.owner] -= 1;
#L150: _emittedCount += 1;
#L163: _numberOfValidTokens[owner] += 1;
#L180: _holdersCount -= 1;
#L184: _numberOfValidTokens[token.owner] -= 1;
can be optimized.

Recommendation

Instead of += 1 or -= 1, use – or ++ prefix to save some operation gas.

For example: --_numberOfValidTokens[token.owner]; would save gas but will do the operation you intended to do.

Alleviation

NTT-05 | Unusual Contract Name

Category	Severity	Location	Status
Coding Style	● Discussion	https://app.bunzz.dev/module-templates/5c7bc93c-b2b7-4d11-9be7-fd62fa9a22b1/code/TokenERC4671.sol	ⓘ Pending

Description

Contract TokenERC4671 has different naming style than other files - IERC4671.sol, IERC4671Enumerale.sol, and IERC4671Metadata.sol

Recommendation

I suggest to change contract file name from TokenERC4671.sol to ERC4671.sol as well as contract name of the file.

Alleviation

Appendix

Finding Categories

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.

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.

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.