SOLIDIFIED

Audit Report for Animoca - October 12, 2021

Summary

Audit Report prepared by Solidified covering a subset of the Animoca smart contracts.

Process and Delivery

Three (3) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief on 21 September 2021.

Audited Files

The source code has been supplied in the form of specific commits in GitHub repositories:

https://github.com/animoca/ethereum-contracts-assets/tree/c999ebac8bf5a2e7df3273363cf13f74ab9e2dba/contracts/token/ERC721

https://github.com/animoca/ethereum-contracts-assets/tree/c999ebac8bf5a2e7df3273363cf13f74ab9e2dba/contracts/token/ERC1155

https://github.com/animoca/ethereum-contracts-assets/tree/c999ebac8bf5a2e7df3273363cf13f74ab9e2dba/contracts/token/ERC1155721

https://github.com/animoca/ethereum-contracts-assets/tree/c999ebac8bf5a2e7df3273363cf13f74ab9e2dba/contracts/mocks/token/ERC721

https://github.com/animoca/ethereum-contracts-assets/tree/c999ebac8bf5a2e7df3273363cf13f74ab9e2dba/contracts/mocks/token/ERC1155

https://github.com/animoca/ethereum-contracts-assets/tree/c999ebac8bf5a2e7df3273363cf13f74ab9e2dba/contracts/mocks/token/ERC1155721

https://github.com/animoca/revv-ethereum-contracts/blob/0a8000542296a71c5e78567428d213088a530678/contracts/token/ERC155721/REVVMotorsportInventory.sol

https://github.com/animoca/tokenlaunchpad-ethereum-contracts/blob/fa2ca40e35e41308058812894fa0b543d3b577bd/contracts/token/ERC1155/TokenLaunchpadVouchers.sol

UPDATE:
Fixes have provided tand the final commit number covered by this report are::

https://github.com/animoca/ethereum-contracts-assets/tree/c97194714ba362d02db667e066b9884c7f94ee05

https://github.com/animoca/tokenlaunchpad-ethereum-contracts/tree/74ae703a23c374016df7312a5221b36505f2ba8d

https://github.com/animoca/revv-ethereum-contracts/tree/19847aa91aaae846287ebe6977e10529da57b087

Intended Behavior

The smart contracts implement ERC-721 and ERC-1155 token implementations and specific instances of the ERC-1155 token used for a motorsport implementation.

Code Complexity and Test Coverage

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note, that high complexity or lower test coverage does equate to a higher risk. Certain bugs are more easily detected in unit testing than a security audit and vice versa. It is, therefore, more likely that undetected issues remain if the test coverage is low or non-existent.

Criteria	Status	Comment
Code complexity	Medium	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	High	-

Issues Found

Solidified found that the Animoca contracts contain no critical issues, no major issues, 1 minor issues, in addition to 2 informational notes.

We recommend all issues are amended, while the notes are up to the team's discretion, as they refer to best practices.

Issue #	Description	Severity	Status
1	ERC1155TokenReceiverMock.sol and ERC721ReceiverMock.sol: Anyone can trigger token Receive events	Minor	Resolved
2	Multiple contracts: Minting is allowed while the contract is paused	Note	Resolved
3	ERC721.sol & ERC1155721Inventory.sol: Allows setting the approval bit without an actual approval	Note	Resolved
4	Misc notes		

Critical Issues

No critical issues have been found.

Major Issues

No major issues have been found.

Minor Issues

1. `ERC1155TokenReceiverMock.sol` and `ERC721ReceiverMock.sol`: Anyone can trigger token Receive events

Anyone can call functions `onERC1155Received()`, `onERC1155BatchReceived()` and `onERC721Received()` and trigger token received events without actually transferring the tokens.
Additionally, the `Received`, `ReceivedSingle` and `ReceivedBatch` events do not contain the `msg.sender` information in the event data.

Recommendation
Consider restricting the callers of the `onERC1155Received()`, `onERC1155BatchReceived()` and `onERC721Received()` functions to a whitelisted set of trusted tokens contracts.

Notes

2. Multiple contracts: Minting is allowed while the contract is paused

The contracts `REVVMotorsportInventory.sol`, `TokenLaunchpadVouchers.sol`, `ERC1155InventoryPausableMock.sol`, `ERC1155721InventoryPausableMock.sol` and `ERC721PausableMock.sol` allows minting tokens while the contract is in paused state.

Recommendation
Consider assessing if it is intentional.

3. ERC721.sol & ERC1155721Inventory.sol: Allows setting the approval bit without an actual approval

The function `approve()` allows to set approval for a zero address. This results in setting the approval bit without changing the value in the approval mapping.

Recommendation
Consider adding a zero address validation to the approve method.

4. Misc notes

1. `ERC721.sol` - function `_batchMint()` writes but never reads the local `values` array.
2. `PausableCollections.sol` - contract is not used in the codebase

Disclaimer

Solidified audit is not a security warranty, investment advice, or an endorsement of Animoca or its products. This audit does not provide a security or correctness guarantee of the audited smart contract. Securing smart contracts is a multistep process, therefore running a bug bounty program as a complement to this audit is strongly recommended.

The individual audit reports are anonymized and combined during a debrief process, in order to provide an unbiased delivery and protect the auditors of Solidified platform from legal and financial liability.

Solidified Technologies Inc.