Arcadeum - Security Audit

Table of content

Introduction

The Horizon team requested the review of some smart contracts on the repository https://github.com/horizon-games/SkyWeaver-contracts, the commit referenced for this audit is `5422f3b56750df1d6b603d7d1319cfbee1c744d7` (https://github.com/horizon-games/SkyWeaver-contracts/commit/5422f3b56750df1d6b603d7d1319cfbee1c744d7).

The audited contracts are:

- **TieredOwnable.sol:** (https://github.com/horizon-games/SkyWeaver-contracts/blob/5422f3b56750df1d6b603d7d1319cfbee1c744d7/contracts/utils/TieredOwnable.sol) Assign ownership tiers to addresses, allowing inheriting contracts to choose which tier can call which function.
- **BridgeFactory.sol:** (https://github.com/horizon-games/SkyWeaver-contracts/blob/5422f3b56750df1d6b603d7d1319cfbee1c744d7/contracts/factories/BridgeFactory.sol) This is a contract allowing contract owner to mint up to N assets per 6 hours. Anyone can send SW assets or ARC to this contract, which will then get burned for the former and reserved for Horizon for the latter. Most of the logic for what to mint and what is considered a valid burn is kept off-chain, in a L2 network.
- **FreemintFactory.sol:** (https://github.com/horizon-games/SkyWeaver-contracts/blob/5422f3b56750df1d6b603d7d1319cfbee1c744d7/contracts/factories/FreemintFactory.sol) This is a contract allowing the owner to mint any tokens within a given range. This factory will be used to mint community-related assets, special event assets that are meant to be given away.
- **Conquest.sol:** (https://github.com/horizon-games/SkyWeaver-contracts/blob/5422f3b56750df1d6b603d7d1319cfbee1c744d7/contracts/POA/Conquest.sol) Contract used on POA to internally keep track of players participation in Conquest.
- **ConquestEntriesFactory.sol:** (https://github.com/horizon-games/SkyWeaver-contracts/blob/5422f3b56750df1d6b603d7d1319cfbee1c744d7/contracts/POA/ConquestEntriesFactory.sol) Contract used on POA allowing players to convert their silver cards to conquest entries. This contract should only be able to mint conquest entries.

The contracts are well written.

The rest of the contracts in the repository are assumed to be audited along with the contracts imported from different repositories as *`multi-token-standard`*.

TieredOwnable.sol

Notes

- N1 - line 2 - pragma experimental ABIEncoderV2 seems to not being used.
- N2 - line 38 - Possible loss of Highest owners. The function *`assignOwnership`* allows senders with the highest tier to change the *`ownerTier`*. By a human mistake, in a scenario where there is only one master owner, that master can change its own tier value, and therefore the function will be unreachable as the @dev notation says. Consider checking that the sender can not change its own tier, using a counter of masters, and/or add a method to explicit renounce to the tier.

BridgeFactory.sol

To take into consideration

- *`onERC1155Received`* and *`onERC1155BatchReceived`* are using the owner of the asset to assign the deposit (*`from`*) but in some cases, the operators could perform the action and be the recipient of the conquer tickets. Consider using *`_data`* with the recipient.
- *`PERIOD_LENGTH`* could be upgradeable.
- *`bachMint`* can be called by a user with the lowest tier value possible, `1`. If for some reason one of that address is compromised, a bigger amount and/or other ids than the burned previously could be passed as a parameter. Consider using an event to *retry* the mint instead of *refund* the tokens. Both scenarios will have concerns but maybe a retry is a better UX.

FreemintFactory.sol

Notes

- N1 - line 72 - Typo `[…] support.s`.
- N1 - line 76 - *`supportsInterface`* can use *`type(IERC165).interfaceId`* as the other contracts to avoid copy/paste's human errors.

To take into consideration

- line 43 - `bachMint` could receive 2 dimensions array for `_ids` and `_amounts` to allow different ids and amounts per beneficiary.

Conquest.sol

To take into consideration

- If conquest entries will start using decimals, consider changing the line 117 `require(_amounts[0] == 1, "Conquest#entry: INVALID_ENTRY_TOKEN_AMOUNT");` where `1` should be `100`.

ConquestEntriesFactory.sol

Low Severity

- L1 - line 126 - Possible loss of cards. If a user sends amounts with decimals and the final number tickets to be minted has decimals (not a natural number), those decimals will be removed, and the user will lose them. Consider using tickets with decimal or check if the amount has decimal on each receive.

Extra

While the audit, a low severity bug was found at the following contracts: `ERC1155Meta.sol` (https://github.com/arcadeum/multi-token-standard/blob/master/contracts/tokens/ERC1155/ERC1155Meta.sol) and `ERC1155MetaPackedBalance.sol` (https://github.com/arcadeum/multi-token-standard/blob/master/contracts/tokens/ERC1155PackedBalance/ERC1155MetaPackedBalance.sol) at the functions `metaSafeTransferFrom` and `metaSafeBatchTransferFrom` where if `_isGasFee` is true, the data passed to the on receive hook (`_callonERC1155BatchReceived`) is `signedData` instead of `transferData` which is not the expected behavior. The Horizon team fixed it (https://github.com/arcadeum/multi-token-standard/pull/59) before this audit was published.

Ignacio Mazzara - 08/2020