HAECHI AUDIT

SuperRare

Smart Contract Security Analysis

Published on : Nov 16, 2021

Version 2.0

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

HAECHI AUDIT

Smart Contract Audit Certificate

SuperRare

Security Report Published by HAECHI AUDIT
v1.0 Nov 10, 2021
v2.0 Nov 16, 2021

Auditor : Felix Kim

Executive Summary

Severity of Issues	Findings	Resolved	Unresolved	Acknowledged	Comment
Critical	-	-	-	-	-
Major	-	-	-	-	-
Minor	1	1	-	-	All Issues Resolved
Tips	-	-	-	-	-

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

TABLE OF CONTENTS

1 Issues (0 Critical, 0 Major, 1 Minor) Found

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

ABOUT US

HAECHI AUDIT believes in the power of cryptocurrency and the next paradigm it will bring. We have the vision to *empower the next generation of finance*. By providing security and trust in the blockchain industry, we dream of a world where everyone has easy access to blockchain technology.

HAECHI AUDIT is a flagship service of HAECHI LABS, the leader of the global blockchain industry. HAECHI AUDIT provides specialized and professional smart contract security auditing and development services.

We are a team of experts with years of experience in the blockchain field and have been trusted by 300+ project groups. Our notable partners include Universe,1inch, Klaytn, Badger, etc.

HAECHI AUDIT is the only blockchain technology company selected for the Samsung Electronics Startup Incubation Program in recognition of our expertise. We have also received technology grants from the Ethereum Foundation and Ethereum Community Fund.

Inquiries: audit@haechi.io

Website: audit.haechi.io

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

INTRODUCTION

This report was prepared to audit the security of bazaar smart contract created by SuperRare team. HAECHI AUDIT focused on whether the smart contract created by SuperRare team is soundly implemented and designed as specified in the published materials, in addition to the safety and security of the smart contract.

CRITICAL	Critical issues must be resolved as critical flaws that can harm a wide range of users.
MAJOR	Major issues require correction because they either have security problems or are implemented not as intended.
MINOR	Minor issues can potentially cause problems and therefore require correction.
TIPS	Tips issues can improve the code usability or efficiency when corrected.

HAECHI AUDIT recommends SuperRare team improve all issues discovered. The following issue explanation uses the format of {file name}#{line number}, {contract name}#{function/variable name} to specify the code. For instance, *Sample.sol:20* points to the 20th line of Sample.sol file, while *Sample#fallback()* means the fallback() function of the Sample contract. Please refer to the Appendix to check all results of the tests conducted for this report.

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

SUMMARY

The codes used in this Audit can be found at GitHub (https://github.com/HAECHI-LABS/audit-superRare/tree/569c874c3d246ec772a8798b86bf2e7587ce5bac). The last commit of the code used in this Audit is "569c874c3d246ec772a8798b86bf2e7587ce5bac".

Issues	HAECHI AUDIT found 0 critical issues, 0 major issues, and 1 minor issues. There are 0 tips issues that can improve the code's usability or efficiency upon modification.
Update	[v.2.0] In the new commit da948edf732a7bdb0530ad43b20297ff09ad2462, 1 minor issue has been corrected.

Severity	Issue	Status
MINOR	SuperRareBazaarBase#_checkSplits() function is not called consistently.	(Found - v1.0) (Resolved - v2.0)

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

OVERVIEW

Contracts subject to audit

- ISuperRareBazaar
- SuperRareBazaar
- SuperRareBazaarBase
- ISuperRareAuctionHouse
- SuperRareAuctionHouse
- ISuperRareMarketplace
- SuperRareMarketplace
- SuperRareBazaarStorage

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

FINDINGS

MINOR

SuperRareBazaarBase#_checkSplits() function is not called consistently.

(Found - v.1.0) (Resolved - v.2.0)

```
51    /// @notice Verifies that the splits supplied are valid.
52    /// @dev A valid split has the same number of splits and ratios.
53    /// @dev There can only be a max of 5 parties split with.
54    /// @dev Total of the ratios should be 100 which is relative.
55    /// @param _splits The addresses the amount is being split with.
56    /// @param _ratios The ratios each address in _splits is getting.
57    function _checkSplits(address[] calldata _splits, uint8[] calldata _ratios)
58        internal
59        pure
60    {
61        require(_splits.length > 0, "checkSplits::Must have at least 1 split");
62        require(_splits.length <= 5, "checkSplits::Split exceeded max size");
63        require(
64            _splits.length == _ratios.length,
65            "checkSplits::Splits and ratios must be equal"
66        );
67        uint256 totalRatio = 0;
68
69        for (uint256 i = 0; i < _ratios.length; i++) {
70            totalRatio += _ratios[i];
71        }
72
73        require(totalRatio == 100, "checkSplits::Total must be equal to 100");
74    }
```

[https://github.com/HAECHI-LABS/audit-superRare/blob/569c874c3d246ec772a8798b86bf2e7587ce5bac/contracts/SuperRareBazaarBase.sol#L51-L74]

Issue

SuperRareBazaarBase#_checkSplits() function receives the addresses to distribute the amount and the ratio of the distribution as parameters, and confirms whether the two parameters are normal. Among the confirming statements exists a require() statement that confirms whether there are 5 or less addresses for distributing the amount.

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

However, even when there are 4 functions in total that receive the addresses for and ratio of distribution as parameters, namely *SuperRareBazaar#setSalePrice(), SuperRareBazaar#accpeptOffer(), SuperRareBazaar#configureAuction(), SuperRareBazaar#convertOfferToAuction()*, only the *SuperRareMarketplace#setSalePrice()* function calls the *SuperRareBazaarBase#_checkSplits()* function and confirms whether there are 5 or less addresses that would receive the distribution. Thus, from functions except *SuperRareMarketplace#setSalePrice()*, the amount can be distributed to over 5 addresses.

Recommendation

We recommend adding a statement that calls the *SuperRareBazaarBase#_checkSplits()* function in the functions *SuperRareBazaar#accpeptOffer(), SuperRareBazaar#configureAuction(), SuperRareBazaar#convertOfferToAuction()*.

Update

[v2.0] - The issue has been resolved by adding a code calling *SuperRareBazaarBase#_checkSplits()* function to *SuperRareBazaar#accpeptOffer(), SuperRareBazaar#configureAuction(), SuperRareBazaar#convertOfferToAuction()*.

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

DISCLAIMER

This report does not guarantee investment advice, the suitability of the business models, and codes that are secure without bugs. This report shall only be used to discuss known technical issues. Other than the issues described in this report, undiscovered issues may exist such as defects on the Main Network. In order to write secure smart contracts, correction of discovered problems and sufficient testing thereof are required.

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

Appendix A. Test Results

The following results show the unit test results covering the key logic of the smart contract subject to the security audit. Parts marked in red are test cases that failed to pass the test due to existing issues.

```
SuperRareBazaar
  Offer Process
    #offer()
      ✓ should fail if offer amount is zero
      ✓ should fail if offer amount less than previous offer + minimum increase value (71ms)
      ✓ should fail if msg.sender does not approve currency to bazaar contract
      ✓ should fail if msg.value is not zero when not using eth (42ms)
      ✓ should fail if msg.sender does not have enough balance
      ✓ should fail if msg.sender is token owner (38ms)
      valid case
        ✓ bazaar takes required amount of token from the msg.sender
        ✓ if previous offer exist, refund currency to buyer
        ✓ should emit OfferPlaced event
    #cancleOffer()
      ✓ should fail if no offer for currency exists
      ✓ should fail if msg.sender does not place current offer (40ms)
      valid case
        ✓ user takes token amount of offer
        ✓ should emit CancelOffer event
    #acceptOffer()
      ✓ should fail if msg.sender is not token owner
      ✓ should fail if nft owner does not approve bazaar
      ✓ should fail if no offer exists
      ✓ should fail if offer amount mismatch (54ms)
      valid case
          ✓ beneficiaries takes commission (55ms)
          ✓ splitted addresses takes fee w.r.t ratio (59ms)
          ✓ should emit AcceptOffer event (55ms)
  Target Sale Process
    #setSalePrice()
      ✓ should fail if msg.sender is not token owner
      ✓ should fail if nft owner does not approve bazaar
      ✓ should fail if split addresses and ratios length mismatch
      ✓ should fail if split addresses length is zero
      ✓ should fail if split addresses length greater than five
      ✓ should fail if sum of split ratios is not 100
      valid case
        ✓ sets a sale price for the given nft directed at the target address
        ✓ should emit SetSalePrice event
```

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

```
  #removeSalePrice()
    ✓ should fail if msg.sender is not token owner
    valid case
      ✓ target sale information removed
      ✓ should emit SetSalePrice event
  #buy()
    ✓ should fail if nft owner does not approve bazaar
    ✓ should fail if token is not registered for sale
    ✓ should fail if price setter and current nft owner mismatch (66ms)
    ✓ should fail if insufficient amount parameter given (51ms)
    ✓ should fail if buyer does not have enough currency (64ms)
    valid case
      ✓ beneficiaries takes commission (67ms)
      ✓ splitted addresses takes fee w.r.t ratio (71ms)
      ✓ if buyer has offer, refund (66ms)
      ✓ nft marked as sold
      ✓ should emit Sold event (65ms)
Auction Process
  #configureAuction()
    ✓ should fail if msg.sender is not token owner
    ✓ should fail if token owner does not approve bazaar contract
    ✓ should fail if auction is too long
    ✓ should fail if already auction start
    ✓ should fail if length of auction is zero
    ✓ should fail if coldie auction price is zero
    ✓ should fail if scheduled auction start time set to past
    ✓ should fail if starting amount is greater than max value
    valid case
      ✓ auction registered (75ms)
      ✓ if auction is scheduled auction, target nft transferred to bazaar (67ms)
      ✓ should emit NewAuction event (68ms)
  #convertofferToAuction()
    ✓ should fail if msg.sender is not token owner
    ✓ should fail if token owner does not approve bazaar contract
    ✓ should fail if auction already exists (51ms)
    ✓ should fail if auction is too long
    ✓ should fail if try to convert offer with different amount
    valid case
      ✓ coldie auction registered (44ms)
      ✓ offer information removed
      ✓ should emit NewAuction event (38ms)
      ✓ should emit AuctionBid event (39ms)
  #cancelAuction()
    ✓ should fail if auction is not configured
    ✓ should fail if auction is already started
    ✓ should fail if msg.sender is not creator nor owner
    valid case
      ✓ auction information removed (39ms)
      ✓ token transferred auction creator if bazaar has token
```

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

```
    ✓ should emit CancelAuction event
  #bid()
    ✓ should fail if msg.sender dost not have enough currency approval
    ✓ should fail if auction is not configured
    ✓ should fail if auction creator try to bid
    ✓ should fail if auction is not started
    ✓ should fail if currency mismatch (47ms)
    ✓ should fail if try to bid zero amount
    ✓ should fail if try to bid more than max value
    ✓ should fail if try to bid less than min value
    ✓ should fail if schedule auction is ended
    ✓ should fail if try to be less than previous bid + min increase (54ms)
   valid case - with multiple bids
     ✓ bid information updates properly (98ms)
  #settleAuction()
    ✓ should fail if there is no valid auction
    ✓ should fail if auction is not ended
     valid case
       ✓ token transferred to highest bidder (94ms)
       ✓ bid amount distributed to beneficiaries
       ✓ should emit AuctionSettled event (89ms)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
contracts/bazaar					
ISuperRareBazaar.sol	100	100	100	100	
SuperRareBazaar.sol	100	100	100	100	
SuperRareBazaarBase.sol	100	100	100	100	
contracts/auctionhoust					
ISuperRareAuctionHouse.sol	100	100	100	100	
SuperRareAuctionHouse.sol	100	100	100	100	
contracts/marketplace					
ISuperRareMarketplace.sol	100	100	100	100	
SuperRareMarketplace.sol	100	100	100	100	
contracts/storage					

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

SuperRareBazaarStorage.sol	100	100	100	100	

[Table 1] Test Case Coverage

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved

End of Document

COPYRIGHT 2021. HAECHI AUDIT. all rights reserved