AUDIT REPORT

BackBone Labs

Necropolis Contracts

Prepared by SCV-Security

On 27rd June 2024

Table of Contents

Introduction

SCV has been engaged by BackBone Labs to conduct a comprehensive security review with the goal of identifying potential security threats and vulnerabilities within the codebase. The purpose of this audit is to evaluate the security posture of the codebase and provide actionable recommendations to mitigate any identified risks. This report presents an overview of the findings from our security audit, outlining areas of concern and proposing effective measures to enhance the codebase's security.

Scope Functionality

The protocol allows NFT owners to sell their NFT with support for creators' royalties. Sellers can receive payments by accepting a direct bidder offer, performing auction bidding, or letting a buyer purchase them directly.

Submitted Codebase

bbl-necropolis-contracts	
Repository	https://github.com/backbonelabsio/bbl-necropolis-contracts
Commit	`fbad5f572c5d3de0c4a96abb38b257d8d13e0c3b`
Branch	`master`

Revisions Codebase

bbl-necropolis-contracts	
Repository	https://github.com/backbonelabsio/bbl-necropolis-contracts
Commit	`ac5b9fc33b54a1506bf786c7240e1a9df4b489c2`
Branch	`audit_revision`

Methodologies

SCV performs a combination of automated and manual security testing based on the scope of testing. The testing performed is based on the extensive experience and knowledge of the auditor to provide the greatest coverage and value to BackBone Labs. Testing includes, but is not limited to, the following:

- Understanding the application and its functionality purpose.
- Deploying SCV in-house tooling to automate dependency analysis and static code review.
- Analyse each line of the code base and inspect application security perimeter.
- Review underlying infrastructure technologies and supply chain security posture.

Code Criteria

This section provides an evaluation of specific criteria aspects as described below:

- **Documentation:** Evaluating the presence and comprehensiveness of publicly available or provided explanatory information, diagram flowcharts, comments, and supporting documents to enhance code understanding.
- **Coverage:** Evaluating whether the code adequately addresses all necessary cases and scenarios, ensuring that the intended functionality or requirements are sufficiently covered.
- **Readability:** Assessing how easily the code can be understood and maintained, considering factors such as code structure, naming conventions, and overall organisation.
- **Complexity:** Evaluating the complexity of the code, including factors such as, number of lines, conditional statements, and nested structures.

The status of each criteria is categorised as either **SUFFICIENT** or **NOT-SUFFICIENT** based on the audit assessment. This categorisation provides insights to identify areas that may require further attention and improvement.

Criteria	Status	Notes
Documentation	**SUFFICIENT**	N/A
Coverage	**SUFFICIENT**	`cargo tarpaulin` reports a coverage of 71.67%.
Readability	**SUFFICIENT**	N/A
Complexity	**NOT-SUFFICIENT**	Several TODO comments with unfinished implementations are in the codebase, indicating a potential increase in complexity in the future.

Findings Summary

Summary Title	Risk Impact	Status
Auctions with CW20 tokens cannot be finalized	CRITICAL	PARTIALLY RESOLVED
Settling auction may fail due to an out-of-gas limit	CRITICAL	RESOLVED
Offers are not removed after refunding, allowing users to get extra funds	CRITICAL	PARTIALLY RESOLVED
Canceling auctions or accepting offers does not refund the current bid	CRITICAL	RESOLVED
Accepting offers does not transfer NFT and send funds	CRITICAL	RESOLVED
Users can bid without providing funds	CRITICAL	RESOLVED
Depositing with CW20 tokens causes loss of funds	CRITICAL	RESOLVED
Bidders can make offers during frozen states and settled auctions	MODERATE	RESOLVED
Incorrect user balance check when making offers	MODERATE	RESOLVED
Protocol fee configuration is used instead of the auction's configuration	MODERATE	RESOLVED
Inaccessible code	INFO	RESOLVED

Findings Technical Details

1. Auctions with CW20 tokens cannot be finalized

RISK IMPACT: CRITICAL	STATUS: PARTIALLY RESOLVED

Description

The `settle_auction` function in `contracts/auction/src/auction.rs:497` determines whether the auction denom is a CW20 token by checking whether the token starts with "*cw20:*". This is incorrect because a CW20 token auction denom will be denominated as their contract address.

As a result, auctions using CW20 tokens will fail to be settled, causing sellers and bidders to lose funds.

Recommendation

Consider storing the token type (native denom or CW20 token) when creating the auction and use it to determine the kind of tokens when settling it.

2. Settling auction may fail due to an out-of-gas limit

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

When settling and canceling an auction, all of the auction's offers are refunded in `contracts/auction/src/auction.rs:516`, `contracts/auction/src/auction.rs:527`, and `contracts/auction/src/auction.rs:799`. Since the functions iterate over all the offers, an attacker can create many offers with zero amounts to force the transaction to fail due to an out-of-gas error.

As a result, the auction cannot be finalized, causing sellers and bidders to lose funds.

Recommendation

Consider modifying the implementation so users must revoke the offers manually instead of refunding them automatically.

3. Offers are not removed after refunding, allowing users to get extra funds

RISK IMPACT: CRITICAL	STATUS: PARTIALLY RESOLVED

Description

When settling and canceling an auction, all of the auction's offers are refunded in `contracts/auction/src/auction.rs:516`, `contracts/auction/src/auction.rs:527`, and `contracts/auction/src/auction.rs:799` by increasing the `USER_BALANCES` state. However, the refunded offers have not been removed from the auction state (`AUCTIONS.offers`).

This allows users who have received the refunds to receive another refund via the `revoke_offer` function in `contracts/auction/src/auction.rs:1073`.

As a result, users can bid without providing funds, allowing them to steal funds originating from other bidders.

Recommendation

Consider setting `auction.offers` to `None` after performing all the offer refunds.

Additionally, the `TODO` comment in `contracts/auction/src/auction.rs:956-963` should be resolved by ensuring users can only make a maximum amount of one offer. This is required to ensure users who perform multiple bids can receive all their refunds in `contracts/auction/src/auction.rs:517-520`, as the current implementation does not refund them. The old `OFFERS_BY_BIDDER` state in `contracts/auction/src/auction.rs:990-994` will also be overwritten if the bidder makes more than one offer.

4. Canceling auctions or accepting offers does not refund the current bid

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

The `_cancel_auction` function in `contracts/auction/src/auction.rs:785` and the `accept_offer` function in `contracts/auction/src/auction.rs:1026-1028` do not perform a refund to current bidders if they have already placed bids in `contracts/auction/src/auction.rs:295-296` and `contracts/auction/src/auction.rs:312-315`.

As a result, the current bidders will lose funds.

Recommendation

Consider refunding the current bidder when calling the `_cancel_auction` and `accept_offer` functions. If an existing bidder for an auction exists, consider removing the relevant state mappings like `contracts/auction/src/auction.rs:641-653`.

5. Accepting offers does not transfer NFT and send funds

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

The `accept_offer` function in `contracts/auction/src/auction.rs:1032` calls the `settle_auction` function to accept an existing offer from a bidder. However, the messages that must be dispatched from the `settle_auction` function in `contracts/auction/src/auction.rs:535-633` are not called to the response in `contracts/auction/src/auction.rs:1036`.

As a result, the funds are not sent to the fee collector, royalty creator, and seller. The bidder will also lose funds as they will not receive the NFT.

Recommendation

Consider adding `settle_response.messages` to the response in `contracts/auction/src/auction.rs:1036`.

6. Users can bid without providing funds

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

The `PlaceBid` message in `contracts/auction/src/contract.rs:65-68` allows users to specify any form of `cw_balance` and bids into an auction. If the user provides `Balance::Cw20` as any CW20 token address, users can place bids without providing the required funds.

As a result, users can bid with other users' funds and receive the NFT for free.

Recommendation

Consider removing the `Balance::Cw20` entry point and letting users place bids with CW20 tokens via `contracts/auction/src/contract.rs:253-256`.

7. Depositing with CW20 tokens causes loss of funds

RISK IMPACT: CRITICAL	STATUS: RESOLVED

Description

The `ReceiveMsg::Deposit` message in `contracts/auction/src/contract.rs:257-266` deposits the funds to `USER_BALANCES`. This is incorrect because when `ExecuteMsg::Receive` is called, the user's funds are already transferred to the auction contract.

However, the `TransferFrom` message in `contracts/auction/src/auction.rs:916-920` will take the user's deposit again, causing users to send funds twice the amount required to the auction contract but only credited once.

Recommendation

Consider removing the `ReceiveMsg::Deposit` entry point, as users can deposit via `contracts/auction/src/contract.rs:92` with the CW20 allowance mechanism.

8. Bidders can make offers during frozen states and settled auctions

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

The `make_offer` function in `contracts/auction/src/auction.rs:937`, no validation ensures the contract state is not frozen, and the auction is not settled. This means bidders can make offers for settled auctions, but more importantly if the admin freezes the contract, bidders can still make offers.

Recommendation

Consider returning an error if the contract state is frozen or if the auction is settled.

9. Incorrect user balance check when making offers

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

The `make_offer` function in `contracts/auction/src/auction.rs:969` returns the error if the user balance from the `USER_BALANCES` state is lesser than the sum of the existing bid and bid amount. This is incorrect because the user's balance is already deducted for existing bids, as seen in `contracts/auction/src/auction.rs:1000`.

As a result, users cannot make offers when they have sufficient balance.

Recommendation

Consider removing the validation and implementing the `TODO` comment in `contracts/auction/src/auction.rs:956-963`.

10. Protocol fee configuration is used instead of the auction's configuration

RISK IMPACT: MODERATE	STATUS: RESOLVED

Description

The `settle_auction` function in `contracts/auction/src/auction.rs:502` computes the auction's protocol fee with `config.protocol_fee`. This is incorrect because `auction.protocol_fee` should be used, representing the protocol fee value during the auction creation.

If the admin updates the protocol fee value, sellers will receive a different amount than expected.

Recommendation

Consider changing the `config.protocol_fee` to `auction.protocol_fee`.

11. Inaccessible code

RISK IMPACT: INFORMATIONAL	STATUS: RESOLVED

Description

The code block in `contracts/auction/src/auction.rs:973-978` will only be called if `auction.min_offer_percentage` is `None`. This code cannot be entered as auctions can only be created with `min_offer_percentage` as `None` in `contracts/auction/src/contract.rs:224`.

Additionally, all bids can be the least minimum amount required as long they equal or larger than the `min_offer_amount` in `contracts/auction/src/auction.rs:975-977`, which may be incorrect if the assumption of new bid amount should be larger than existing bids.

Recommendation

Consider finishing the functionality or removing the `min_offer_amount` field.

Document Control

Version	Date	Notes
-	13th May 2024	Security audit commencement date.
0.1	23rd May 2024	Initial report with identified findings delivered.
0.5	24th June 2024	Fixes remediations implemented and reviewed.
1.0	27th June 2024	Audit completed, final report delivered.

Appendices

A. Appendix – Risk assessment methodology

SCV-Security employs a risk assessment methodology to evaluate vulnerabilities and identified issues. This approach involves the analysis of both the LIKELIHOOD of a security incident occurring and the potential IMPACT if such an incident were to happen. For each vulnerability, SCV-Security calculates a risk level on a scale of 5 to 1, where 5 denotes the highest likelihood or impact. Consequently, an overall risk level is derived from combining these two factors, resulting in a value from 10 to 1, with 10 signifying the most elevated level of security risk

Risk Level	Range
CRITICAL	10
SEVERE	From 9 to 8
MODERATE	From 7 to 6
LOW	From 5 to 4
INFORMATIONAL	From 3 to 1

LIKELIHOOD and **IMPACT** would be individually assessed based on the below:

Rate	LIKELIHOOD	IMPACT
5	**Extremely Likely**	Could result in severe and irreparable consequences.
4	**Likely**	May lead to substantial impact or loss.
3	**Possible**	Could cause partial impact or loss on a wide scale.
2	**Unlikely**	Might cause temporary disruptions or losses.
1	**Rare**	Could have minimal or negligible impact.

B. Appendix – Report Disclaimer

This report should not be regarded as an "endorsement" or "disapproval" of any specific project or team. These reports do not indicate the economics or value of any "product" or "asset" created by a team or project that engages SCV-Security for a security review. The audit report does not make any statements or warranties about the code's utility, safety, suitability of the business model, regulatory compliance of the business model, or any other claims regarding the fitness of the implementation for its purpose or its bug-free status. The audit documentation is intended for discussion purposes only. The content of this audit report is provided "as is," without representations and warranties of any kind, and SCV-Security disclaims any liability for damages arising from or in connection with this audit report. Copyright of this report remains with SCV-Security.

THANK YOU FOR CHOOSING

SCV SECURITY

scv.services

contact@scv.services