0xCommit

Security Assessment Report

Navix Ecosystem

Navis NFT & Marketplace Audit

Version: Final

Date: 6 Feb 2025

Table of Contents

License

Disclaimer

Introduction

Purpose of this report

0xCommit has been engaged by **Navix Ecosystem** to perform a security audit of several NFT and Marketplace contract components.

The objectives of the audit are as follows:

1. Determine the correct functioning of the protocol, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine smart contract bugs, which might lead to unexpected behaviour.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents a summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebases Submitted for the Audit

The audit has been performed on the following contracts:

https://github.com/Sylas23/Navis-Blockchain

Contracts	Description	Commit hash
- Marin - NavisNFT - NavisMarketplace	Initial Commit	06c679c8165bfe24eda2545abd5d1c174d5e9a58
	Final Commit	cbe54de5cdf6e1e68f6d090c1689cf5980c7c15f

How to Read This Report

This report classifies the issues found into the following severity categories:

Severity	Description
Critical	A serious and exploitable vulnerability that can lead to loss of funds, unrecoverable locked funds, or catastrophic denial of service.
High	An attacker can successfully execute an attack that clearly results in operational issues for the service. This also includes any value loss of unclaimed funds permanently or temporary.
Medium	The service may be susceptible to an attacker carrying out an unintentional action, which could potentially disrupt its operation. Nonetheless, certain limitations exist that make it difficult for the attack to be successful.
Low	The service may be vulnerable to an attacker executing an unintended action, but the impact of the action is negligible or the likelihood of the attack succeeding is very low and there is no loss of value.
Informational	Comments and recommendations of design decisions or potential optimizations, that are not relevant to security. Their application may improve aspects, such as user experience or readability, but is not strictly necessary

The status of an issue can be one of the following: **Pending**, **Acknowledged**, or **Resolved**.

Note that audits are an important step to improving 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 the 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**. We include a table with these criteria below.

Note that high complexity or low test coverage does not necessarily equate to a higher risk, although certain bugs are more easily detected in unit testing than in a security audit and vice versa.

Overview

Methodology

The audit has been performed in the following steps:

1. Gaining an understanding of the code base's intended purpose by reading the available documentation.
2. Automated source code and dependency analysis.
3. Manual line by line analysis of the source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    a. Race condition analysis
    b. Under-/overflow issues
    c. Key management vulnerabilities
    d. Access Control Issues
    e. Boundary Analysis
4. Report preparation

Summary of Findings

Sr. No.	Description	Severity	Status
1	Highest Bidder is not refunded when unlisted a token	Critical	Resolved
2	Owner can frontrun and increase listing price before users buy token	Critical	Resolved
3	Missing check for blacklisted users on listToken function	High	Resolved
4	Allow the highest bidder to conclude the auction as well	Medium	Resolved
5	Incorrect logic around the NFT URI	Low	Resolved
6	The userToNFT mapping in NavisNFT is not updated on transfers	Informational	Resolved

Detailed Findings

Critical Findings

1. Highest Bidder is not refunded when unlisted a token

Severity: Critical

Description

The bidder locks their Navis token into the contract when placing a bid. And when concluding the auction they are transferred the new NFT token. But when the NFT is unlisted by the listing owner, the highest bidder is not refunded.

Impact

- Attackers can create listing and then unlist, thus griefing users of the Navis tokens

Remediation

Refund the highest bidder in the unlist function.

Status

Resolved

2. Owner can frontrun and increase listing price before users buy token

Severity: Critical

Description

The listing owners can update the price of their listing by using the updateListing function. But they can take advantage of this by frontrunning users buying the tokens with/without an auction and can increase the price.

Remediation

Either have a delay in price update logic or do not allow price increase.

Status

Resolved

High Findings

3. Missing check for blacklisted users on listToken function

Severity: High

Description

There is no check for blacklisted users on the listToken function. But it is present on the batchListTokens and batchUnlistTokens functions.

Impact

- Blacklisted users will be able to list new tokens in the marketplace.

Remediation

Use the isNotBlacklisted modifier.

Status

Resolved

Medium Findings

4. Allow the highest bidder to conclude the auction as well

Severity: **Medium**

Description

Currently only the owner of the listing or the Marketplace can conclude the auction. This means if none of them conclude the auction, then funds of the highest bidder are stuck indefinitely.

Remediation

We recommend allowing the highest bidder to conclude the auction.

Status

Resolved

Low Findings

5. Incorrect logic around the NFT URI

Severity: Low

Description

We see some confusing logic of updating the NFT URI at every mint. This is not according to the standard practices. This logic needs to be reworked.

Remediation

Do not update the URI at every mint.

Status

Resolved

Informational Findings

6. The userToNFT mapping in NavisNFT is not updated on transfers

Severity: **Informational**

Description

The userToNFT mapping corresponds to the NFT token IDs held by the user but this does not consider the token transfers because this mapping is only updated when there is a mint of new NFT.

Although we don't see any use of the mapping in the contract, but it might be useful to the UI or Web2 systems.

Remediation

Consider updating this mapping on every transfer.

Status

Resolved