BURRA SEC

Brahma Security Review

Reviewed by: windhustler

24th April, 2025

Brahma Security Review

Burra Security

April 28, 2025

Introduction

A time-boxed security review of the **Brahma** protocol was done by **Burra Security** team, focusing on the security aspects of the smart contracts.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource, and expertise-bound effort where we try to find as many vulnerabilities as possible. We can not guarantee 100% security after the review or even if the review will find any vulnerabilities. Subsequent security reviews, bug bounty programs, and on-chain monitoring are recommended.

About Burra Security

Burra Sec offers security auditing and advisory services with a special focus on cross-chain and inter-operability protocols and their integrations.

About BrahNFT

BrahNFT is a cross-chain NFT implementation built by Brahma.fi using LayerZero's ONFT721 standard. This project enables NFTs to be bridged across different blockchain networks while maintaining ownership integrity.

Severity classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

Impact - The technical, economic, and reputation damage from a successful attack

Likelihood - The chance that a particular vulnerability gets discovered and exploited

Severity - The overall criticality of the risk

Informational - Findings in this category are recommended changes for improving the structure, usability, and overall effectiveness of the system.

Security Assessment Summary

***review commit hash* - c8d00676015063a1a11e0a626d3abce93d99caab**

Scope

The following smart contracts were in the scope of the audit:

- BrahNFT.sol

Findings Summary

ID	Title	Severity	Status
L-01	Use `_safeMint` instead of `_mint`	Low	Resolved
L-02	NFTs can be minted on multiple chains	Low	Acknowledged
L-03	Lack of support for smart contract wallets and LayerZero's composing functionality	Low	Resolved

ID	Title	Severity	Status
I-01	Missing zero address checks	Info	Resolved

Detailed Findings

[L-01] Use `_safeMint` instead of `_mint`

Target

- BrahNFT.sol#L89

Severity

- Impact: Low
- Likelihood: Low

Description

The `BrahNFT::mint` function uses `_mint` instead of `_safeMint` when minting NFTs. This can lead to tokens being permanently locked if minted to a contract address that can't handle NFTs.

Recommendation

Replace `_mint` with `_safeMint`.

BurraSec

Fixed in 00ee20ce2fbdd109c16dcc9add188395255f323d.

[L-02] NFTs can be minted on multiple chains

Target

- BrahNFT.sol#L84

Severity

- Impact: High
- Likelihood: Low

Description

The `BrahNFT` contract implements a signature-based minting mechanism and is designed to be deployed across multiple blockchain networks.

If the `mintValidator` mistakenly provides valid signatures for the same `tokenId` and `toAddress` combination across different chains, it creates a scenario where the same NFT (same `tokenId`) will exist on Chain B, C, etc.

Recommendation

Consider allowing the minting of NFTs only on a single hub chain, where it can be bridged to other chains.

BurraSec

Issue was acknowledged.

[L-03] Lack of support for smart contract wallets and LayerZero's composing functionality

Target

- BrahNFT.sol#L124

Severity

- Impact: Low
- Likelihood: Low

Description

The `BrahNFT::send` function mandates that the receiver of the NFT on the destination chain is the same address as the sender on the source chain. This creates two issues:

1. Smart contract wallets often have different addresses across chains. Forcing the same address can result in tokens being sent to addresses they don't control, leading to loss of funds.
2. LayerZero's composing functionality requires the destination address to implement `lzCompose`. By mandating that the receiver address is the same as `msg.sender`, this functionality is restricted.

Recommendation

Consider allowing an arbitrary `to` address in the `SendParam` struct parameter.

Client

At Brahma accounts, users can create the same smart account with the same address deterministically across several EVMs on which we wanna support the bridging functionality.

BurraSec

68945598aef0490c2bb15f288fa140415a487cb1 disables sending composed messages.

[I-01] Missing zero address checks

Target

- BrahNFT.sol#L69
- BrahNFT.sol#L142

Severity

Informational.

Description

The `BrahNFT` contract lacks zero address validations in both the constructor for `_lzEndpoint`, `_governance`, `_mintValidator` parameters, and in the `setMintValidator` function.

Recommendation

Add zero address validation in the constructor and `setMintValidator` function.

BurraSec

Fixed in 733def86085b45e48d7e2fa7155ffaf0cf899b4e.