shieldify

Onchain Heroes

Genesis Rings Bridge

SECURITY REVIEW

Date: 25 August 2025

CONTENTS

1. About Shieldify

Positioned as the first hybrid Web3 Security company, Shieldify shakes things up with a unique subscription-based auditing model that entitles the customer to unlimited audits within its duration, as well as top-notch service quality thanks to a disruptive 6-layered security approach. The company works with very well-established researchers in the space and has secured multiple millions in TVL across protocols, also can audit codebases written in Solidity, Vyper, Rust, Cairo, Move and Go.

Learn more about us at shieldify.org.

2. Disclaimer

This security review does not guarantee bulletproof protection against a hack or exploit. Smart contracts are a novel technological feat with many known and unknown risks. The protocol, which this report is intended for, indemnifies Shieldify Security against any responsibility for any misbehavior, bugs, or exploits affecting the audited code during any part of the project's life cycle. It is also pivotal to acknowledge that modifications made to the audited code, including fixes for the issues described in this report, may introduce new problems and necessitate additional auditing.

3. About Onchain Heroes - Genesis Rings Bridge

Onchain Heroes is a fully on-chain, idle RPG with novel on-chain game mechanics.

Introduction to Season 2

The next chapter in the ongoing battle for the multiverse. Picking up where Season 1 left off, Season 2 intensifies the fight as our heroes face an unprecedented threat: Dragma, the World Boss. With new mechanics, deeper strategy, and fresh rewards, this season introduces players to an evolved and expanded gameplay experience.

Season 2 Overview

- Progression Reset: All heroes have been reset to Level 1, providing a fresh start for all players.
- Hero Staking: Deploy your heroes into battle to earn $HERO tokens by defeating enemies and progressing through the season.
- New Damage System: Damage is now determined by a combination of hero level and weapon rarity, directly impacting player rewards.
- Weapon Traits: Weapons now feature Sharpness and Durability, which influence their effective damage output and how often they can be used in battle.
- Two-Phase Gameplay:
  - Phase 1: Available at launch, players will face Dragma's Underlings, preparing for the greater challenge ahead.
  - Phase 2: Unlocked several weeks after launch, this phase will bring the direct confrontation with Dragma and higher-stakes gameplay.
- Season Duration: Season 2 will run for approximately 8 to 12 weeks, offering ongoing challenges and evolving gameplay.

Genesis Rings

Genesis Rings are ERC-721 NFTs on the Ethereum Mainnet, representing unique and valuable assets within the Onchain Heroes universe.

Ringbearers are the VIPs of Onchain Heroes, enjoying exclusive in-game perks and special community benefits.

6.9% of the total $HERO Token supply is allocated to Genesis Rings, distributed progressively across seasons.

Additional airdrops and allocations are planned for Ringbearers as the Onchain Heroes universe expands.

4. Risk Classification

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

4.1 Impact

- **High** - results in a significant risk for the protocol's overall well-being. Affects all or most users
- **Medium** - results in a non-critical risk for the protocol affects all or only a subset of users, but is still unacceptable
- **Low** - losses will be limited but bearable - and covers vectors similar to griefing attacks that can be easily repaired

4.2 Likelihood

- **High** - almost certain to happen and highly lucrative for execution by malicious actors
- **Medium** - still relatively likely, although only conditionally possible
- **Low** - requires a unique set of circumstances and poses non-lucrative cost-of-execution to rewards ratio for the actor

5. Security Review Summary

The security review lasted 3 days, with a total of 24 hours dedicated by the Shieldify team.

Overall, the code is well-written. The audit report flagged two Medium severity issues regarding unvalidated native value and potential ETH loss in the `bridgeRingToAbstract()` function.

The Onchain Heroes team has been highly responsive to the Shieldify research team's inquiries and promptly implemented the recommendations.

5.1 Protocol Summary

Project Name	**Onchain Heroes - Genesis Rings Bridge**
Repository	och-bridge-contracts
Type of Project	ERC-721,Abstract Bridge Using LayerZero
Audit Timeline	3 days
Review Commit Hash	6b6bb6bcbf718649fe9ea3539414149f2a6d28aa
Fixes Review Commit Hash	1c86fe64bd3963816b1db1753775452991a125a9

5.2 Scope

The following smart contracts were in the scope of the security review:

File	**nSLOC**
contracts/AbstractBridge.sol	23
contracts/ETHBridge.sol	39
Total	**62**

6. Findings Summary

The following issues have been identified, sorted by their severity:

- **Medium** issues: 2

ID	**Title**	**Severity**	**Status**
[M-01]	User Can Lose ETH in `bridgeRingToAbstract()`	Medium	Fixed
[M-02]	Executor Can Deliver Cross-Chain Messages with Unvalidated Native Value	Medium	Acknowledged

7. Findings

[M-01] User Can Lose ETH in `bridgeRingToAbstract()`

Severity

Medium Risk

Description

The vulnerability arises in the `bridgeRingToAbstract()` function of the `ETHBridge` contract. The function begins by checking whether the provided `tokenIds` array is empty. If the length of the array is zero, the function immediately returns without performing any further operations. However, the function is marked payable, and no refund logic is implemented prior to returning. As a result, if a caller mistakenly sends a non-zero `msg.value` while providing an empty tokenIds array, the transaction will succeed, but the ETH sent will remain trapped within the contract. Since the contract does not expose any withdrawal mechanism for the owner or users, this ETH cannot be recovered and will remain permanently locked.

Location of Affected Code

File: contracts/ETHBridge.sol#L69

```
function bridgeRingToAbstract(address recipient, uint256[] calldata
   tokenIds, bytes calldata options) external payable {
    uint256 len = tokenIds.length;

    // If `tokenIds.length == 0`, then return function
    if (len == 0) {
        return;
    }
    // code
}
```

Impact

This issue creates a condition where user funds can be silently lost without any observable indication of an error. A user intending to bridge NFTs could mistakenly send ETH alongside an empty tokenIds array. Because the transaction does not revert and the ETH remains in the contract, users may incur direct financial loss.

Recommendation

To prevent ETH from being trapped, the function should enforce a non-empty tokenIds array as a precondition. Instead of silently returning when the array length is zero, the function should revert with a clear error message, ensuring that no ETH is accepted in such cases.

Team Response

Fixed.

[M-02] Executor Can Deliver Cross-Chain Messages with Unvalidated Native Value

Severity

Medium Risk

Description

The `AbstractBridge` contract contains a vulnerability in its `_lzReceive()` function, where it fails to validate the `msg.value` parameter against the intended native value that should accompany cross-chain message execution. This vulnerability stems from LayerZero's execution model, where any party can execute verified messages, and the receiving contract cannot guarantee that the execution parameters match those originally specified in the executor options.

The vulnerability manifests in the `_lzReceive()` function within `AbstractBridge.sol`, which processes incoming cross-chain messages without implementing native value validation. The current message decoding logic `abi.decode(_message, (address, uint256[]))` only extracts the recipient address and token IDs, completely omitting any validation of the native value that should accompany the message execution.

When users initiate bridging operations through the ETHBridge contract's `bridgeRingToAbstract()` function, they specify executor options that may include native value requirements for destination chain execution. However, the `AbstractBridge` contract operates under the assumption that the `msg.value` received during `_lzReceive()` execution matches the originally intended amount.

Reference - enforce-msgvalue-in-_lzreceive-and-lzcompose

Location of Affected Code

File: contracts/AbstractBridge.sol#L39

```
function _lzReceive(Origin calldata, bytes32, bytes calldata _message,
   address, bytes calldata) internal override {
    // code
    (address recipient, uint256[] memory tokenIds) = abi.decode(_message,
        (address, uint256[]));

    uint256 len = tokenIds.length;
    for (uint256 i = 0; i < len; ++i) {
    // Transfer the ring to this contract
        OCH_RING_CONTRACT.transferFrom(address(this), recipient, tokenIds
            [i]);
    }
}
```

Impact

While the current Genesis Ring bridge implementation does not explicitly utilize msg.value in its processing logic but it is best practice to check `msg.value`.

Recommendation

To address this vulnerability, the bridge contracts must implement comprehensive validation of native value parameters by encoding intended execution values in cross-chain messages and validating them during destination execution.

Team Response

Acknowledged.

our shielding · Your smart contracts, our shielding · Your smart c

shieldify

Thank you!