Contract Audit Results

Prepared on: Jun 09, 2021

Contract: AUD436

Prepared by:

Charles Holtzkampf

Sentnlio Ltd

Prepared for:

Nilotpal Mukherjee

Genesis Shards

Table of Contents

Executive Summary

This document outlines any issues found during the audit of the contracts:

- genshards

The contract has 0 flaws or security vulnerabilities. The risk associated with this contract is low risk

REMARK	MINOR	MAJOR	CRITICAL
0	0	0	0

Severity Description

REMARK

Remarks are instances in the code that are worthy of attention, but in no way represent a security flaw in the code. These issues might cause problems with the user experience, confusion with new developers working on the project, or other inconveniences.

Things that would fall under remarks would include:

- Instances where best practices are not followed
- Spelling and grammar mistakes
- Inconsistencies in the code styling and structure

MINOR

Issues of Minor severity can cause problems in the code, but would not cause the code to crash unexpectedly or for funds to be lost. It might cause results that would be unexpected by users, or minor disruptions in operations. Minor problems are prone to become major problems if not addressed appropriately.

Things that would fall under minor would include:

- Logic flaws (excluding those that cause crashes or loss of funds)
- Code duplication
- Ambiguous code

MAJOR

Issues of major security can cause the code to crash unexpectedly, or lead to deadlock situations.

Things that would fall under major would include:

- Logic flaws that cause crashes
- Timeout exceptions
- Incorrect ABI file generation
- Unrestricted resource usage (for example, users can lock all RAM on contract)

CRITICAL

Critical issues cause a loss of funds or severely impact contract usage.

Things that would fall under critical would include:

- Missing checks for authorization
- Logic flaws that cause loss of funds
- Logic flaws that impact economics of system
- All known exploits (for example, on_notification fake transfer exploit)

Methodology

Throughout the review process, we check that the token contract:

- Documentation and code comments match logic and behaviour
- Is not affected by any known vulnerabilities

Our team follows best practices and industry-standard techniques to verify the proper implementation of the smart contract. Our smart contract developers reviewed the contract line by line, documenting any issues as they were discovered.

Our strategies consist largely of manual collaboration between multiple team members at each stage of the review, including:

I. Due diligence in assessing the overall code quality of the codebase.
II. Testing contract logic against common and uncommon attack vectors.
III. Thorough, manual review of the codebase, line-by-line.

Our testing includes

- Overflow Audit
- Authority Control Audit Authority Vulnerability Audit
- Authority Excessive Audit
- Safety Design Audit Hard-coded Audit
- Show coding Audit
- Abnormal check Audit
- Type safety Audit
- Denial of Service Audit
- Performance Optimization Audit
- Design Logic Audit
- False Notice Audit
- False Error Notification Audit

- Counterfeit Token Audit
- Random Number Security Audit
- Rollback Attack Audit

Contract Files

Filename	SHA256
GenFactory.sol	899f8a676ab9bc704df4d230c7cddcd35dca443faa2c7acfa821bfffd5117bd4
GenShards.sol	48c33aa157e18ab434f08178448da20ecf44a4ddbb3d0bc54cde798141ca7d9a
GenAccess.sol	a0c891d14c6c3352f5de4f51f045ee59a47461a9a2b5beb2b3a06cc62a134503
IGenMarketFactory.sol	8e2d38b4e16c2cda967c05ec80c3ed9e1272cbc11eca33cfb34972172cd6ce3c
GenMarket.sol	9973e3b0db98f0e94653771269e8a23a6eaa9211949e67c5719a76d8f7581ec1
GenMarketFactory.sol	ff1ad54dff6a8c7f7866249b40c01fb4be5df201994cfc0510ef804065446575
GenTickets.sol	93f3592f714ddb2c39353c0cbfa01704d40c5345100c82869de93ef69d461db4