Goldon Studios

Audit Report

contact@movebit.xyz

https://twitter.com/movebit_

Wed Nov 22 2023

Goldon Studios Audit Report

1 Executive Summary

1.1 Project Information

Description	A FPS AAA Shooter Game
Type	NFT
Auditors	MoveBit
Timeline	Tue Nov 21 2023 - Wed Nov 22 2023
Languages	Move
Platform	Sui
Methods	Architecture Review, Unit Testing, Manual Review
Source Code	https://github.com/shubhoum/RageEffect-x-Sui
Commits	a653014393038eaa1e747542cabc004a3727840e

1.2 Files in Scope

The following are the SHA1 hashes of the original reviewed files.

ID	File	SHA-1 Hash
RAG	Smart contract/rage.move	46daf71456e597e483ca81995d0cf63b69454db0

1.3 Issue Statistic

Item	Count	Fixed	Acknowledged
Total	2	0	2
Informational	0	0	0
Minor	1	0	1
Medium	1	0	1
Major	0	0	0
Critical	0	0	0

1.4 MoveBit Audit Breakdown

MoveBit aims to assess repositories for security-related issues, code quality, and compliance with specifications and best practices. Possible issues our team looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Integer overflow/underflow by bit operations
- Number of rounding errors
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting
- Unchecked CALL Return Values
- The flow of capability
- Witness Type

1.5 Methodology

The security team adopted the **"Testing and Automated Analysis"**, **"Code Review"** and **"Formal Verification"** strategy to perform a complete security test on the code in a way that is closest to the real attack. The main entrance and scope of security testing are stated in the conventions in the "Audit Objective", which can expand to contexts beyond the scope according to the actual testing needs. The main types of this security audit include:

(1) Testing and Automated Analysis

Items to check: state consistency / failure rollback / unit testing / value overflows / parameter verification / unhandled errors / boundary checking / coding specifications.

(2) Code Review

The code scope is illustrated in section 1.2.

(3) Formal Verification

Perform formal verification for key functions with the Move Prover.

(4) Audit Process

- Carry out relevant security tests on the testnet or the mainnet;
- If there are any questions during the audit process, communicate with the code owner in time. The code owners should actively cooperate (this might include providing the latest stable source code, relevant deployment scripts or methods, transaction signature scripts, exchange docking schemes, etc.);
- The necessary information during the audit process will be well documented for both the audit team and the code owner in a timely manner.

2 Summary

This report has been commissioned by Goldon Studios to identify any potential issues and vulnerabilities in the source code of the Goldon Studios smart contract, as well as any contract dependencies that were not part of an officially recognized library. In this audit, we have utilized various techniques, including manual code review and static analysis, to identify potential vulnerabilities and security issues.

During the audit, we identified 2 issues of varying severity, listed below.

ID	Title	Severity	Status
RAG-1	Centralization Risk	Medium	Acknowledged
RAG-2	Missing Third-Party Dependency	Minor	Acknowledged

3 Participant Process

Here are the relevant actors with their respective abilities within the Goldon Studios Smart Contract:

Admin

- Admin can mint `Rage` object for a receiver through `mint` function.

4 Findings

RAG-1 Centralization Risk

Severity: Medium

Status: Acknowledged

Code Location:

Smart contract/rage.move#107

Descriptions:

There is a risk of centralization in the contract, the admin can call the call `mint` function at will to get the `Rage` object.

Suggestion:

It is recommended to take some measures to mitigate centralization risk.

Resolution:

The client replied that they will be using multi-sigs to avoid over-centralization.

RAG-2 Missing Third-Party Dependency

Severity: Minor

Status: Acknowledged

Code Location:

Smart contract/rage.move

Descriptions:

During the audit process, we discovered that the system relies on some third-party services for certain functionalities, such as the `nft_protocol` module. However, please note that this audit does not cover the third-party dependencies, including the `ob_permissions`, `ob_request`, and so on. We assume that the function provided by those modules is safe.

Suggestion:

It is recommended to utilize audited and widely adopted third-party dependencies whenever possible. Necessary security measures should be implemented to address potential issues that may arise from these dependencies. Additionally, proactive monitoring of the third-party services is essential during the operational phase to promptly detect and mitigate any potential risks and avoid potential losses.

Resolution:

According to the reply of the client, the third-party dependencies are from OriginByte, and they have undergone an audit.

Appendix 1

Issue Level

- **Informational** issues are often recommendations to improve the style of the code or to optimize code that does not affect the overall functionality.
- **Minor** issues are general suggestions relevant to best practices and readability. They don't post any direct risk. Developers are encouraged to fix them.
- **Medium** issues are non-exploitable problems and not security vulnerabilities. They should be fixed unless there is a specific reason not to.
- **Major** issues are security vulnerabilities. They put a portion of users' sensitive information at risk, and often are not directly exploitable. All major issues should be fixed.
- **Critical** issues are directly exploitable security vulnerabilities. They put users' sensitive information at risk. All critical issues should be fixed.

Issue Status

- **Fixed:** The issue has been resolved.
- **Partially Fixed:** The issue has been partially resolved.
- **Acknowledged:** The issue has been acknowledged by the code owner, and the code owner confirms it's as designed, and decides to keep it.

Appendix 2

Disclaimer

This report is based on the scope of materials and documents provided, with a limited review at the time provided. Results may not be complete and do not include all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your own risk. A report does not imply an endorsement of any particular project or team, nor does it guarantee its security. These reports should not be relied upon in any way by any third party, including for the purpose of making any decision to buy or sell products, services, or any other assets. TO THE FULLEST EXTENT PERMITTED BY LAW, WE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, IN CONNECTION WITH THIS REPORT, ITS CONTENT, RELATED SERVICES AND PRODUCTS, AND YOUR USE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NOT INFRINGEMENT.