QuillAudits

Audit Report
September, 2022

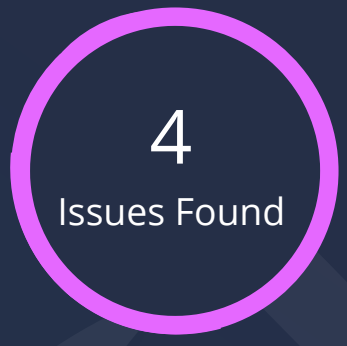
For

GAMELOFT

Table of Content

Executive Summary

Project Name	AsphaltNFT
Overview	AsphaltNFT is an ERC721 smart contract which will be deployed on Ethereum L1 blockchain to integrate with L2 NFTs on ImmutableX.
Timeline	May 16, 2022 - July 11, 2022
Method	Manual Review, Functional Testing, Automated Testing etc.
Scope of Audit	The scope of this audit was to analyse AsphaltNFT smart contract for quality, security, and correctness. *https://github.com/gameloft/glana/* **Commit hash:** b5d5181a8e5cc749ae7123cac2ea8e0785feec9f **Branch:** develop



High
Medium
Low
Informational

	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	0	4
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	0	0	0

Types of Severities

High

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

Checked Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- Exception Disorder
- Gasless Send
- Use of tx.origin
- Compiler version not fixed
- Address hardcoded
- Divide before multiply
- Integer overflow/underflow
- Dangerous strict equalities
- Tautology or contradiction
- Return values of low-level calls
- Missing Zero Address Validation
- Private modifier
- Revert/require functions
- Using block.timestamp
- Multiple Sends
- Using SHA3
- Using suicide
- Using throw
- Using inline assembly

Techniques and Methods

Throughout the audit of smart contract, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

Static analysis of smart contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis.

Manual Testing

A. Contract - AsphaltNFT

High Severity Issues

No issues found

Medium Severity Issues

No issues found

Low Severity Issues

No issues found

Informational Issues

A.1: Unindexed event parameters

Description

Events do not have any indexed parameters. Unindexed parameters make it difficult to track important data for off-chain monitoring tools.

Remediation

Consider indexing event parameters to avoid the task of off-chain services searching and filtering for specific events.

Status

Acknowledged

Gameloft team's Comment: These events belong to functions that are not being called regularly, so we just leave them as that to save gas.

A.2 Redundant code

Description

mintFor() function takes quantity as parameter which is getting checked on L59 which expects it to be 1. But this quantity parameter is not getting used in this function which makes quantity and the require check redundant.

Remediation

Consider removing redundant code.

Status

Acknowledged

Gameloft team's Comment: This parameter is required by IMX to mint NFTs, so we have to keep it.

A.3 Centrailization

Description

malicious owner can mint NFTs to any addresses using mintFor() or can change base uri using setBaseURI() or change imx address using setImx() at any time, this is necesarry that an authorized address should have access to these functionalities but centrailization can be avoided here by using multisig wallet.

Remediation

Consider transferring ownership to multisig wallet.

Status

Acknowledged

Gameloft team's Comment: We might consider transfer the contract's ownership to a multisig wallet later.

A.4 General recommendation

Description

mintFor() is using _safeMint() to mint NFTs which can be used by receiver contract to reenter into transaction flow if this function is getting called in any other contract which lacks the required checks for stopping reentrancy.

Ref: When “SafeMint” Becomes Unsafe

Remediation

Care needs to be taken while making calls to this function from other contracts as described in description above.

Status

Acknowledged

Functional Testing

- Owner should be able to set imx address using setImx
- Owner should be able to enable or disable NFT minting
- Owner or approved address should be able to burn the token
- Owner should be able to set base uri
- Only owner or imx address should be able to mint nfts
- Reverts if caller is not owner while enabling minting with setEnableImx
- Reverts if caller is not imx adress and not owner address while minting nfts with mintFor
- Reverts if unauthorized account tries to set base url with setBaseURI

Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of the AsphaltNFT . We performed our audit according to the procedure described above.

Some issues of informational severity were found, Some suggestions and best practices are also provided in order to improve the code quality and security posture.

Disclaimer

QuillAudits smart contract audit is not a security warranty, investment advice, or an endorsement of the AsphaltNFT Platform. This audit does not provide a security or correctness guarantee of the audited smart contracts.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the AsphaltNFT Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties..

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

600+
Audits Completed

$15B
Secured

600K
Lines of Code Audited

Follow Our Journey

Audit Report
September, 2022

For

GAMELOFT

QuillAudits

Canada, India, Singapore, United Kingdom

audits.quillhash.com

audits@quillhash.com