META LEGEND APES

Security Audit

March 20th,2023

V 1.0.0

Prepared by
Mayank Meena
(fiverr.com/mayankmeena)

Introduction

This document includes the results of the security audit for smart contract (0xC8F5969e50125B73C8CF77738c61896cb4c6A025 provided by Alan21000 on fiverr)
code as found in the section titled ‘Source Code’. The security audit was performed by the Mayank’s security team from Mar 17, 2023 to Mar 21, 2023. The purpose of this audit is to review the source code of certain Solidity contracts, and provide feedback on the design, architecture, and quality of the source code with an emphasis on validating the correctness and security of the software in its entirety.

Disclaimer:
While Mayank’s team review is comprehensive and has surfaced some changes that should be made to the source code, this audit should not solely be relied upon for security, as no single audit is guaranteed to catch all possible bugs.

Overall Assessment
We identified a few issues of low to high severity.

Specification
Our understanding of the specification was based on the following sources:
- Discussions on Fiverr with the Alan21000
- The official website, Source code found on blockchain.

Additional Info about Audited Project

Name - META LEGEND APES
Symbol - MLA
Website - https://www.metalegendapes.xyz
OpenSea -https://opensea.io/collection/meta-legend-apes
Total Supply - 5000
Minted/ claimed - 01/5000 as of 20/03/2023

Defined claim condition

start date 17-03-2023 3:11 pm
drop size - 50 NFTs
Mint Price - 0.025 weth
per wallet Limit - 10

Roles

Admin - 1. 0xaD0467D741d65E3954248fEa502E900fb6C76ff9
2. 0x85E6C2B9c8E24Daa0E0F85CD97266b2Fe5964f74

Creator - 0xaD0467D741d65E3954248fEa502E900fb6C76ff9

Owner - 0x85e6c2b9c8e24daa0e0f85cd97266b2fe5964f74

Source Code

the following source code was reviewed during the audit:

Link :- [Source Code]

Note: This document contains an audit solely of the Solidity contracts listed above. Specifically, the audit pertains only to the contracts themselves, and does not pertain to any other programs or scripts, including deployment scripts and libraries.

Library used in Contract

IERC721AUpgradeable.sol IERC165Upgradeable.sol ERC165Upgradeable.sol StringsUpgradeable.sol MulticallUpgradeable.sol ContextUpgradeable.sol AddressUpgradeable.sol IERC721MetadataUpgradeable.sol IERC721Upgradeable.sol IERC721ReceiverUpgradeable.sol Initializable.sol IERC2981Upgradeable.sol SafeERC20.sol ERC2771ContextUpgradeable.sol TWStrings.sol TWAddress.sol	MerkleProof.sol CurrencyTransferLib.sol IWETH.sol IRoyalty.sol IPrimarySale.sol IPlatformFee.sol IPermissionsEnumerable.sol IPermissions.sol IOwnable.sol IOperatorFilterToggle.sol IOperatorFilterRegistry.sol ILazyMint.sol IDrop.sol IDelayedReveal.sol IContractMetadata.sol IClaimConditionMultiPhase.sol IClaimCondition.sol Royalty.sol PrimarySale.sol	OperatorFilterToggle.sol LazyMint.sol Drop.sol DelayedReveal.sol DefaultOperatorFiltererUpgradeable.sol ContractMetadata.sol BatchMintMetadata.sol IERC2981.sol IERC20.sol IERC165.sol ERC721AVirtualApproveUpgradeable.sol PlatformFee.sol PermissionsEnumerable.sol Permissions.sol Ownable.sol OperatorFiltererUpgradeable.sol DropERC721.sol

Methodology

The audit was conducted in several steps.

First, we reviewed in detail all available documentation and specifications for the project, as described in the ‘Specification’ section above.

Second, we performed a thorough manual review of the code, checking that the code matched up with the specification, as well as the spirit of the contract (i.e. the intended behavior). During this manual review portion of the audit we primarily searched for security vulnerabilities, unwanted behavior vulnerabilities, and problems with systems of incentives.

Third, we performed the automated portion of the review consisting of measuring test coverage (while also assessing the quality of the test suite) and evaluating the results of various symbolic execution tools against the code.

Lastly, we performed a final line-by-line inspection of the code – including comments –in effort to find any minor issues with code quality, documentation, or best practices.

Issues Descriptions and Recommendations

Severity Level Reference:

Level	Description
High	The issue poses existential risk to the project, and the issue identified could lead to massive financial or reputational repercussions. We highly recommend fixing the reported issue. If you have already deployed, you should upgrade or redeploy your contracts.
Medium	The potential risk is large, but there is some ambiguity surrounding whether or not the issue would practically manifest. We recommend considering a fix for the reported issue.
Low	The risk is small, unlikely, or not relevant to the project in a meaningful way. Whether or not the project wants to develop a fix is up to the goals and needs of the project.
Code Quality	The issue identified does not pose any obvious risk, but fixing it would improve overall code quality, conform to recommended best practices, and perhaps lead to fewer development issues in the future.
Gas Optimizations	The presented optimization suggestion would save an amount of gas significant enough, in our opinion, to be worth the development cost of implementing it.

Based on this Levels We rate certain part of Smart Contract
In High based on Severity Level Reference, after we have found after checking all parameter and given information by Alan21000.

Report on Smart Contract for META LEGEND APES:

The smart contract for META LEGEND APES appears to be well-written and follows best practices. It includes several features such as Contract Metadata, Platform Fee, Royalty, Primary Sale, Ownable, DelayedReveal, LazyMint, PermissionsEnumerable, and Drop. The SPDX-License-Identifier is also included, indicating the license under which the code is released, and the code uses Solidity version 0.8.11.

The contract defines a claim condition, which starts on March 17th, 2023, at 3:11 pm, and allows for the minting of 50 NFTs at a price of 0.025 WETH per token. There is also a limit of 10 tokens per wallet. The contract includes several roles, such as Admin, Creator, and Owner. The Admin role is held by two Ethereum addresses, 0xaD0467D741d65E3954248fEa502E900fb6C76ff9 and 0x85E6C2B9c8E24Daa0E0F85CD97266b2Fe5964f74. The Creator role is held by 0xaD0467D741d65E3954248fEa502E900fb6C76ff9, and the Owner role is held by 0x85e6c2b9c8e24daa0e0f85cd97266b2fe5964f74.

The contract includes several functions for managing permissions, such as grantRole() and revokeRole(). There are also functions for managing token minting and transfer, such as mintToken() and transferToken(). However, it is worth noting that the contract allows the Admin to change the royalty percentage, wallet, and even add their wallet for additional royalties, which could potentially be a security threat.

The META LEGEND APES NFTs have a name of MLA, a symbol of MLA, and a total supply of 5000. As of March 20th, 2023, only one NFT has been claimed. The NFTs can be viewed on OpenSea at https://opensea.io/collection/meta-legend-apes, and the project's website is https://www.metalegendapes.xyz.

Overall, the contract appears to be well-designed and follows best practices, but a more thorough audit would be necessary to identify any potential vulnerabilities or issues with the contract.

Additionally. there is a potential threat that the admin can change the royalty percentage, wallet, and even add their wallet for additional royalty. This could potentially harm the interests of the other stakeholders, such as the creator and the owner, and may require further investigation.

All Functions of Smart Contract

READ Functions

1. DEFAULT_ADMIN_ROLE - Shows default admin
2. Balance Of - shows nft balance of wallet
3. Claim Condition - minting condition
4. Contract Type - code type of contract via thirdweb
5. Contract URI - details stored on ipfs about contract
6. Contract Version - third web contract version 4
7. Get Active Claim Condition Id - id of mint contion batch
8. Get Approved - Approval of NFT for actions related
9. Get Default Royalty Info - Royalty % info
10. Get Platform Fee Info - additonal royalties set by dev
11. Get RevealURI - NFT metadata URI
12. Max Total Supply - Total supply
13. Name - Name of Contract
14. Owner - Current Owner of Contract
15. Primary Sale Recipient - shows wallet of royaltie reciver
16. Total Supply - minted NFTs

Write Functions

1. Approve - Approve token for trading
2. Burn - Burn token
3. Claim - claim nft minting
4. Grant Role - give certain role to wallet
5. Multicall - called for giving multiple instruction to smart contract
6. Renounce Role - give role to some one else wallet
7. Reveal - NFT reveal
8. Revoke Role - remove role of specific wallet
9. Set Claim Conditions - set minting nft rules
10. Set ContractURI - set settings uri of smart contract
11. Set Default Royalty Info - set Royalties
12. Set Max Total Supply
13. Set Owner - Give owner role to other

Disclaimer

Mayank's team makes no warranties, either express, implied, statutory, or otherwise, with respect to the services or deliverables provided in this report, and Mayank's team specifically disclaims all implied warranties of merchantability, fitness for a particular purpose, noninfringement and those arising from a course of dealing, usage or trade with respect thereto, and all such warranties are hereby excluded to the fullest extent permitted by law. Mayank's team will not be liable for any lost profits, business, contracts, revenue, goodwill, production, anticipated savings, loss of data, or costs of procurement of substitute goods or services or for any claim or demand by any other party. In no event will Mayank's team be liable for consequential, incidental, special, indirect, or exemplary damages arising out of this agreement or any work statement, however caused and (to the fullest extent permitted by law) under any theory of liability (including negligence), even if Mayank's team has been advised of the possibility of such damages. The scope of this report and review is limited to a review of only the code presented by the Emergent team and only the source code Mayank's team notes as being within the scope of Mayank's team's review within this report. This report does not include an audit of the deployment scripts used to deploy the Solidity contracts in the repository corresponding to this audit. Specifically, for the avoidance of doubt, this report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. In this report you may through hypertext or other computer links, gain access to websites operated by persons other than Mayank's team. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such websites' owners. You agree that Mayank's team is not responsible for the content or operation of such websites, and that Mayank's team shall have no liability to your or any other person or entity for the use of third party websites. Mayank's team assumes no responsibility for the use of third party software and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.