Certified
Certified Blockchain Security Professional™

Certified
Certified Solidity Developer™

OVERVIEW

PROJECT SUMMARY

Project	**Animals Meta Club**
Platform	N/a
Language	Solidity

AUDIT SUMMARY

Date	10-04-2022	
Audit Type	Static Analysis, Manual Review	
Audit Result	**PENDING**	
Auditor	**Jarmo van de Seijp**	https://tinyurl.com/Jvdseijp

RISK SUMMARY

Risk Level	Total	Found	Pending	Solved	Acknowledged	Objected
Critical	1	1	1	0	0	0
Major	0	0	0	0	0	0
Medium	1	1	1	0	0	0
Minor	3	3	3	0	0	0
Informative	17	17	17	0	0	0
Discussion	0	0	0	0	0	0

FINDINGS

Access Control

Description:
The owner of the contract is transferred to an arbitrary address without checking if the recipient is able to accept ownership, or is a contract address with no method of controlling the ownership functions. In case of a mistakenly transferred ownership, it would be lost permanently

Category	Risk Level	Number of Findings	Status
Push-Over-Pull	Medium	**1**	Pending

Non-failsafe Whitelist usage

Description:
The current whitelist methodology of creatting a mapping with a list of addresses has a number of known flaws. The writing of addresses will become increasingly expensive as the data writing on a blockchain is inefficient. Writing to, as well as reading from a large whitelist mapping may also hit block limits, which can cause transactions to fail due to the block rejecting them.

Recommendation:
Using OpenZeppelin's MerkleTree contract will ensure a limitless whitelist at the lowest possible datacost (32bytes), and an instant lookup and validation of an address's whitelist status.

Category	Risk Level	Number of Findings	Status
Block Limits	Critical	**1**	Pending

Unused Code

SWC-ID: SWC-131

Relationship:
CWE-1164: Irrelevant Code

Description:
Unused variables are allowed in Solidity and they do not pose a direct security issue. It is best practice though to avoid them as they can:

- cause an increase in computations (and unnecessary gas consumption)

- indicate bugs or malformed data structures and they are generally a sign of poor code quality

- cause code noise and decrease readability of the code

Relevance:

important entire libraries from OpenZeppelin includes a lot of code that the NFT contract actually never uses. This is strictly information as it poses no threat whatsoever

Category	Risk Level	Number of Findings	Status
SWC-131	Informational	**17**	pending

Missing Events

Description:
The contract may change significant state variables in the contract, but does not emit these changes in events. This may result in lack of transparency or 3rd party applications being unable to properly register the contract's current state

Relevance:

setCost and **setMaxSupply, setPresaleCost** should emit an event.

Category	Risk Level	Number of Findings	Status
Missing events	minor	**2**	Pending

Risk Of Centralization

Description:
The owner of the contract has the power to significantly change the economics from within the contract. There are 15 privileged functions controlled by 1 address. A safer method is to assign different roles to privileged functions, based on their level of 'trust'

Category	Risk Level	Number of Findings	Status
Control Flow	Minor	**1**	Pending

AUDIT RESULT

Basic Coding Bugs

1. Constructor Mismatch

o Description: Whether the contract name and its constructor are not identical to each other.

o Result: PASSED

o Severity: Critical

Ownership Takeover

o Description: Whether the set owner function is not protected.

o Result: PASSED

o Severity: Critical

Redundant Fallback Function

o Description: Whether the contract has a redundant fallback function.

o Result: PASSED

o Severity: Critical

Overflows & Underflows

Description: Whether the contract has general overflow or underflow Vulnerabilities

o Result: PASSED

o Severity: Critical

Reentrancy

o Description: Reentrancy is an issue when code can call back into your contract and change state, such as withdrawing ETHs.

o Result: PASSED

o Severity: Critical

MONEY-Giving Bug

o Description: Whether the contract returns funds to an arbitrary address.

o Result: PASSED

o Severity: High

Blackhole

o *Description: Whether the contract locks ETH indefinitely: merely in without out.*

o *Result: PASSED*

o *Severity: High*

Unauthorized Self-Destruct

o *Description: Whether the contract can be killed by any arbitrary address.*

o *Result: PASSED*

o *Severity: Medium*

Revert DoS

o *Description: Whether the contractis vulnerable to DoSattack because of unexpected revert.*

o *Result: PASSED*

o *Severity: Medium*

Unchecked External Call

o *Description: Whether the contract has any external call without checking the return value.*

o *Result: PASSED*

o *Severity: Medium*

Gasless Send

o *Description: Whether the contractis vulnerable to gasless send.*

o *Result: PASSED*

o *Severity: Medium*

Send Instead of Transfer

o *Description: Whether the contract uses send instead of transfer.*

o *Result: PASSED*

o *Severity: Medium*

Costly Loop

o Description: Whether the contract has any costly loop which may lead to Out-Of-Gas exception.

o Result: PASSED

o Severity: Medium

(Unsafe) Use of Untrusted Libraries

o Description: Whether the contract use any suspicious libraries.

o Result: PASSED

o Severity: Medium

(Unsafe) Use of Predictable Variables

o Description: Whether the contract contains any randomness variable, but its value can be predicated.

o Result: PASSED

o Severity: Medium

Transaction Ordering Dependence

o Description: Whether the final state of the contract depends on the order of the transactions.

o Result: PASSED

o Severity: Medium

. Deprecated Uses

o Description: Whether the contract use the deprecated tx.origin to perform the authorization.

o Result: PASSED

o Severity: Medium