Certified

Certified Blockchain Security Professional™

Certified

Certified Solidity Developer™

OVERVIEW

PROJECT SUMMARY

Project	**NFTes**
Platform	Polygon
Language	Solidity

AUDIT SUMMARY

Date	25-05-2022	
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	2	2	2	0	0	0
Medium	2	2	2	0	0	0
Minor	1	1	1	0	0	0
Informative	42	42	42	0	0	0
Discussion	0	0	0	0	0	0

FINDINGS

Function Default Visibility

SWC-ID: SWC-100
Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:
Functions that do not have a function visibility type specified are public by default. This can lead to a vulnerability if a developer forgot to set the visibility and a malicious user is able to make unauthorized or unintended state changes.

Relevance:
cryptoRandomMint, fiatRandoMint,contractBalance, withDraw, checkTotalMinted, getSTatusMintFeeAndMaxMints,setStatusMintFeeAndMaxMints, checkMintedCategoryWise, returnNftsOwner, changeOwner, isAdmin should be declared external

Category	Risk Level	Number of Findings	Status
SWC-100	informational	**11**	Pending

Constable State

Relationship:
CWE-710: Improper Adherence to Coding Standards

Description:

Constant state variables should be declared constant to save gas.

Relevance:

maxDiamondCount, maxGoldCount, MaxMintLimit ,

maxSilverCount should be declared constant

Category	Risk Level	Number of Findings	Status
SWC-108	informational	**4**	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:
function isAdmin is declared as an external view function, but only returns a boolean value indicating whether or not the callee is '*Owner*'. There is no apparent

Category	Risk Level	Number of Findings	Status
SWC-131	Informational	**27**	pending

Note:
a lot of the inherited (26 functions) from libraries contain unused code. I can give you the full list. Since you're imprting from files directly, and not importing from @openzeppelin, it might save bytecode to remove this list as well

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:

setStatusMintFeeAndMaxMints should emit an event.

Category	Risk Level	Number of Findings	Status
Missing events	minor	**1**	Pending

Risk Of Centralization

Description:
The owner of the contract has the power to significantly change the economics from within the contract. All privileged functions are set to the Owner without governance. Losing access to the owner account would result in permanent loss of control of these functions.

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

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	**2**	Pending

Access Control

Description:
The contract sets both an 'owner' and 'Owner', the former being relevant to privileged functions, and the other being only relevant to the redundant function isAdmin, and as part of the pseudo-randomness process. It's highly advisable to remove 'Owner' alltogether to prevent confusion or loss of privilege.

Category	Risk Level	Number of Findings	Status
Ambiguous Owner	Major	**2**	Pending

Bad Randomness

Description:
The contract relies on predictable data to generate the randomness for the NFT's to determine their 'level (diamond, gold etc). This can be easily exploited by a solver contract and should be remidiated before going live.

Relevance:
CWE-330: Use of Insufficiently Random Values

Category	Risk Level	Number of Findings	Status
SWC-120	Critical	**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