KISHIELD

Security Audit

FGD NFTs

May 8, 2022

FGD

Table of Contents

1 **Audit Summary**

2 **Project Overview**

2.1 Token Summary

2.2 Main Contract Assessed

3 **Smart Contract Vulnerability Checks**

4 **Contract Ownership**

4.1 Priviliged Functions

5 **Important Notes To The Users**

6 **Findings Summary**

6.1 Classification of Issues

6.1 Findings Table

01 Code with no effects

02 Code with no effects

03 Code with no effects

04 Variables could be declared as constant

05 Code with no effects

06 Unnecessary counter reset

7 **Statistics**

7.1 Liquidity

7.2 Token Holders

7.3 Liquidity Holders

8 **Liquidity Ownership**

9 **Disclaimer**

Audit Summary

This report has been prepared for FGD NFTs on the Binance Chain network. KISHIELD provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Ensuring contract logic meets the specifications and intentions of the client without exposing the user's funds to risk.
- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Project Overview

Token Summary

Parameter	Result
Address	0x176b537758c15b759699298c3e8fc3e9ece4ba77
Name	FGD NFTs
Platform	Binance Chain
compiler	v0.8.0+commit.c7dfd78e
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/ address/0x176b537758c15b759699298c3e8fc3e9ece4ba77
Url	https://fgd.ai/

Main Contract Assessed

Name	Contract	Live
NFT	0x176b537758c15b759699298c3e8fc3e9ece4ba77	Yes
StakePool	0x8a8da57b532f567cfe2d2d7e411897a04875da18	Yes

Smart Contract Vulnerability Checks

Vulnerability	Automatic Scan	Manual Scan	Result
Unencrypted Private Data On-Chain	Complete	Complete	✓ Low / No Risk
Code With No Effects	Complete	Complete	✓ Low / No Risk
Message call with hardcoded gas amount	Complete	Complete	✓ Low / No Risk
Hash Collisions With Multiple Variable Length Arguments	Complete	Complete	✓ Low / No Risk
Unexpected Ether balance	Complete	Complete	✓ Low / No Risk
Presence of unused variables	Complete	Complete	✓ Low / No Risk
Right-To-Left-Override control character (U+202E)	Complete	Complete	✓ Low / No Risk
Typographical Error	Complete	Complete	✓ Low / No Risk
DoS With Block Gas Limit	Complete	Complete	✓ Low / No Risk
Arbitrary Jump with Function Type Variable	Complete	Complete	✓ Low / No Risk
Insufficient Gas Griefing	Complete	Complete	✓ Low / No Risk
Incorrect Inheritance Order	Complete	Complete	✓ Low / No Risk
Write to Arbitrary Storage Location	Complete	Complete	✓ Low / No Risk
Requirement Violation	Complete	Complete	✓ Low / No Risk
Missing Protection against Signature Replay Attacks	Complete	Complete	✓ Low / No Risk
Weak Sources of Randomness from Chain Attributes	Complete	Complete	✓ Low / No Risk

Vulnerability	Automatic Scan	Manual Scan	Result
Authorization through tx.origin	Complete	Complete	✓ Low / No Risk
Delegatecall to Untrusted Callee	Complete	Complete	✓ Low / No Risk
Use of Deprecated Solidity Functions	Complete	Complete	✓ Low / No Risk
Assert Violation	Complete	Complete	✓ Low / No Risk
Reentrancy	Complete	Complete	✓ Low / No Risk
Unprotected SELFDESTRUCT Instruction	Complete	Complete	✓ Low / No Risk
Unprotected Ether Withdrawal	Complete	Complete	✓ Low / No Risk
Unchecked Call Return Value	Complete	Complete	✓ Low / No Risk
Outdated Compiler Version	Complete	Complete	✓ Low / No Risk
Integer Overflow and Underflow	Complete	Complete	✓ Low / No Risk
Function Default Visibility	Complete	Complete	✓ Low / No Risk

Contract Ownership

The contract ownership of FGD NFTs is not currently renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address 0x10b9d0dee624e079980a42f6d9e01e982c1979af which can be viewed from:
HERE

The owner wallet has the power to call the functions displayed on the priviliged functions chart below, if the owner wallet is compromised this privileges could be exploited.

We recommend the team to renounce ownership at the right timing if possible, or gradually migrate to a timelock with governing functionalities in respect of transparency and safety considerations.

Important Notes To The Users:

- FGD NFTs have 4 tiers all with different supply and prices.
- The total limit of all the tiers at deploy time is 12800 NFTs
- The prices are based on a 18 decimal ERC-20 token.
- The prices for each tier at deploy time are 320, 220, 150, 80 tokens per unit respectively.
- The owner cannot change the prices of the NFTs.
- The amounts for each tier at deploy time are 1200, 2800, 3800, 5000 NFTs per tier respectively.
- Whitelisted users can mint only one 1 tier NFT, if they wish to mint again they must be whitelisted again.
- There can only be 100 NFTs minted by whitelisted users.
- Non-whitelisted users must wait for the owner to set the sale startTime.
- In the case the sale flag is 1 users are required to be validated by the pool Contract and have more than or exactly 5 teammates if they want to mint.
- Once the owner renounces ownership of the contract, none of the following are applicable.
- The owner can change the base URI (URL pointing to artwork).
- The owner can alter the amounts of NFTs for each tier.
- The owner can change the limit of NFTs that an address can mint.
- The owner can change the limit of NFTs that can be minted on a single call.

- The owner can change the start time.
- The owner can add/remove wallets from the whitelist.
- No high-risk Exploits/Vulnerabilities Were Found in token Source Code

Audit Passed

Findings Summary

Classification of Issues

All Issues are of informational.

Severity	Description
🔴 High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency
🟠 Medium	Bugs or issues with that may be subject to exploit, though their impact is somewhat limited.Issues under this classification are recommended to be fixed as soon as possible.
🟡 Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
🟣 Info	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

Findings

Severity	Found
🔴 High	0
🟠 Medium	0
🟡 Low	0
🟣 Info	6
Total	6

Findings

Code with no effects

ID	Severity	Contract	Function
01	● Informational	FGD NFTs	Contract Ownable2.sol

Description

Double use of the Ownable contract. adminaaa is the same address and have the same powers than owner.

Recommendation

We recommend to delete Ownable2.sol and update the onlyMaster modifier. If there is need multiple 'owners' make use of OpenZeppelin Access Control Roles.

Code with no effects

ID	Severity	Contract	Function
02	● Informational	FGD NFTs	function mintWhite()

Description

Unnecessary for-loop. Variable num is set to 1, thus the loop with only execute once.

Recommendation

We recommend allowing whitelisted users to enter the amount of NFTs they want to buy, or deleting the for-loop.

Code with no effects

ID	Severity	Contract	Function
03	● Informational	FGD NFTs	function mintWhite()

Description

mintedCounts[_type] is not updated when whitelisted users mint tier 1 NFTs

Recommendation

We recommend updating the mintedCounts mapping upon minting.

Variables could be declared as constant

ID	Severity	Contract	Function
04	● Informational	FGD NFTs	Variable typeMin, typeMax

Description

Gas Optimization. Variables that are never changed could be declared as constant.

Recommendation

We recommend declaring those variables as constant.

Code with no effects

ID	Severity	Contract	Function
05	● Informational	FGD NFTs	Variables typeMin, typeMax

Description

Variable is not used by the contract.

Recommendation

We recommend deleting or making use of the variable

Unnecessary counter reset

ID	Severity	Contract	Function
06	● Informational	FGD NFTs	idCounter.reset(10001)

Description

Default value of counters in the library Counters is 0

Recommendation

We recommend deleting the counter reset or leave it if this is the planned functionality

Priviliged Functions (onlyMaster)

Function Name	Parameters	Visibility
setBaseURI	string calldata newBaseTokenURI	external
setTypeCount	uint8 _type, uint256 _max	external
setAddressLimit	uint _addressMax	external
setAddressTypeLimit	uint _addressMax	external
setOnceMax	uint _onceMax	external
setStartTime	uint _start	external
setWhiteMax	int256 _max	external
setWhiteList	address[]memory addressList	external
cancelWhiteList	address[]memory addressList	external
setRushSaleFlag	uint256 _flag	external
updateRushSaleTime	uint256 _start, uint _end	external
setRecipientAddress	address _addr	external

Disclaimer

KISHIELD has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided ‘as is’, and KISHIELD is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will KISHIELD or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

The assessment services provided by KISHIELD is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.