CFG NINJA

AUDITS

Security Assessment

Flinch NFT Mint Contract

June 3, 2022

Table of Contents

Audit Summary

This report has been prepared for Flinch NFT Mint Contract on the Ethereum Mainnet network. KronicLabs and CFGNINJA 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 along with the current liquidity and token holder statistics of the protocol.

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:

- 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.

Project Overview

Token Summary

Parameter	Result
Address	0xd4f11C30078d352354c0B17eAA8076C825416493
Name	Flinch NFT
Token Tracker	Flinch NFT (FLN)
Decimals	0
Supply	9999
Platform	Ethereum Mainnet
compiler	v0.8.7+commit.e28d00a7
Contract Name	FlinchNFT
Optimization	Yes with 200 runs
LicenseType	MIT
Language	Solidity
Codebase	https://ropsten.etherscan.io/address/0x6aa731926df284b100c8c2a1696ff892bd049072
Payment Tx	0x02285fb3c4c2774712afc4c4ada5f898f8af3f8e95e94ca248752681a4d4386c
Payment Tx	0x27fa3eef04a4baae69815dac5697612cd203e259e50d5eb267caeea5f83be1b4

Main Contract Assessed
Contract Name

Name	Contract	Live
Flinch NFT	0xd4f11C30078d352354c0B17eAA8076C825416493	No

TestNet Contract Assessed
Contract Name

Name	Contract	Live
Flinch NFT	0x76615FD5892f8a3ccd6274aDCB71b84384CEa9B3	Yes

Solidity Code Provided

SolID	File Sha-1	FileName
Flinch	16c258b7fde29e19f34cd6a0460616fbc1182d13	Flinch.sol
Flinch	6c8ac7f44fb9170c27b5f46cc3fc62723864ef95	ERC721A.sol
Flinch		

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
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

Mint Check

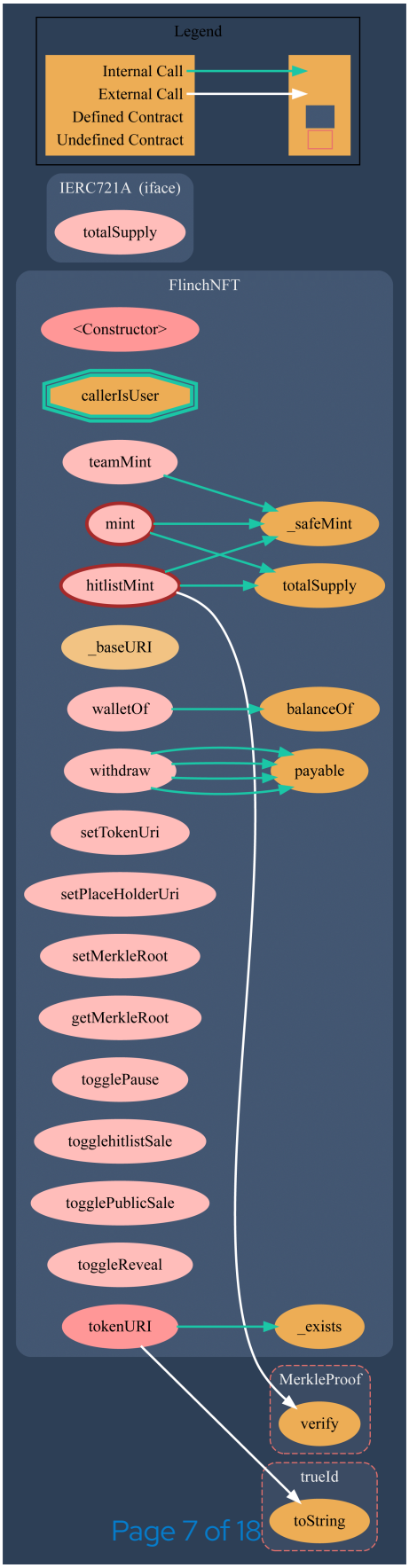
The Project Owners of Flinch NFT has the ability to Mint New ERC 721 tokens.

Since this is an ERC 721 contract is expected to be able to mint new NFTs, so this will be considered a pass for the contract.

Call Graph and Inheritance

The contract for Flinch NFT has the following call graph structure

The Project has a Total Supply of 9999 and has the following inheritance



Contract Ownership

The contract ownership of Flinch NFT 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 0x1333e81c131e1d1d0e8bd42eca5e45acd0ce1de3 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.

We recommend the team to use a Multisignature Wallet if contract is not going to be renounced, this will give the ability to the team to have more control over the contract.

KYC Information

The Project Onwers of Flinch NFT has provided KYC Documentation.

KYC Certificated can be found on the Following:
KYC Data

KYC Information Notes:

Auditor Notes: Auditor asked project owner if there was any plans to KYC.

Project Owner Notes: Customer is Doxxed Cameron Van Hoy https://www.linkedin.com/in/cameron-van-hoy-a51b199/

Mythx Security Summary Checks

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	Flinch.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	Flinch.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	Flinch.sol	L: 0 C: 0
SWC-103	Low	A floating pragma is set.	Flinch.sol	L: 5 C: 0
SWC-104	Pass	Unchecked Call Return Value.	Flinch.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	Flinch.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	Flinch.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	Flinch.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	Flinch.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	Flinch.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	Flinch.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	Flinch.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	Flinch.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	Flinch.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-114	Medium	Transaction Order Dependence.	Flinch.sol	L: 0 C: 0
SWC-115	Low	Authorization through tx.origin.	Flinch.sol	L: 474 C: 15
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	Flinch.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	Flinch.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	Flinch.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	Flinch.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	Flinch.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	Flinch.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	Flinch.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	Flinch.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	Flinch.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	Flinch.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	Flinch.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	Flinch.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	Flinch.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-129	Pass	Typographical Error.	Flinch.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	Flinch.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	Flinch.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	Flinch.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	Flinch.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	Flinch.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	Flinch.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	Flinch.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry standard security scanning tool

Security Check Details Page

SWC-103 - Floating Pragma.

CWE-664: Improper Control of a Resource Through its Lifetime.

Description:

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Remediation:

Lock the pragma version and also consider known bugs (https://github.com/ethereum/solidity/releases) for the compiler version that is chosen.

Pragma statements can be allowed to float when a contract is intended for consumption by other developers, as in the case with contracts in a library or EthPM package. Otherwise, the developer would need to manually update the pragma in order to compile locally.

References:

Ethereum Smart Contract Best Practices - Lock pragmas to specific compiler version.

SWC-114 - Transaction Order Dependence

CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')

Description:

The Ethereum network processes transactions in blocks with new blocks getting confirmed around every 17 seconds. The miners look at transactions they have received and select which transactions to include in a block, based who has paid a high enough gas price to be included. Additionally, when transactions are sent to the Ethereum network they are forwarded to each node for processing. Thus, a person who is running an Ethereum node can tell which transactions are going to occur before they are finalized.A race condition vulnerability occurs when code depends on the order of the transactions submitted to it.

Remediation:

A possible way to remedy for race conditions in submission of information in exchange for a reward is called a commit reveal hash scheme. Instead of submitting the answer the party who has the answer submits hash(salt, address, answer) [salt being some number of their choosing] the contract stores this hash and the sender address. To claim the reward the sender then submits a transaction with the salt, and answer. The contract hashes salt, msg.sender, answer and checks the hash produced against the stored hash, if the hash matches the contract releases the reward.

References:

General Article on Race Conditions ERC20 Race Condition

SWC-115 - Authorization through tx.origin

CWE-477: Use of Obsolete Function

Description:

tx.origin is a global variable in Solidity which returns the address of the account that sent the transaction. Using the variable for authorization could make a contract vulnerable if an authorized account calls into a malicious contract. A call could be made to the vulnerable contract that passes the authorization check since tx.origin returns the original sender of the transaction which in this case is the authorized account.

Remediation:

tx.origin should not be used for authorization. Use msg.sender instead.

References:

Solidity Documentation - tx.origin

Ethereum Smart Contract Best Practices - Avoid using tx.origin

SigmaPrime - Visibility.

SWC Information Notes:

Auditor Notes: Reviewed list of issues with customer.

Project Owner Notes: Customer Acknowledge the issues and answered any concerns from auditor.

Priviliged Functions

Function Name	Parameters	Visibility
teamMint		external
setTokenUri		external
setPlaceHolderUri		external
setMerkleRoot		external
togglePause		external
togglehitlistSale		external
togglePublicSale		external
toggleReveal		public
withdraw		external

Important Notes To The Users:

- The team has been confirmed with KYC and CFG Ninja is has noted them as safe
- Since this is an ERC 721 contract that can a mint, this is needed to generate new NFT. However it has been hardcoded not to go over 10,000 NFTs and the Current Supply is 2,871.
- This is an upgradable contract and this is the proxy:
- https://bscscan.com/address/0x59eFfb5a8caB87f2411EffDD40646B335274AF5E#code
- No high-risk Exploits/Vulnerabilities Were Found in the Source Code.

Audit Passed

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/FlinchNft	Pass
Instagram	https://www.instagram.com/flinchmovie/	Pass
Website	https://www.flinchthemovie.com/	Pass
Discord	https://discord.gg/VKB4xMQdG8	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Disclaimer

KronicLabs and CFGNINJA 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 KronicLabs and CFGNINJA is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will KronicLabs and CFGNINJA 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 KronicLabs and CFGNINJA 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.