Code security assessment

CRYPTOHOTS

CRYPTO
ALERT
NFT

BLOCKCHAIN SECURITY ANALYSIS GROUP

April 2022

v1.2

Index

Synopsis

This audit has been conducted by request of the CRYPTOHOTS project development team to evaluate its smart contracts' code for vulnerabilities or issues, as well as the soundness of its code architecture.

Our analysis unit conducted both automated and manual analysis, including line-by-line examination by our team's experts, checking the whole code for potential vulnerabilities, compiling errors, timestamp / order dependecies, reentrancy errors, known attack vectors and coding best practices.

The contracts were then test-deployed for extensive live testing of all their functionalities including interactions with multiple clients.

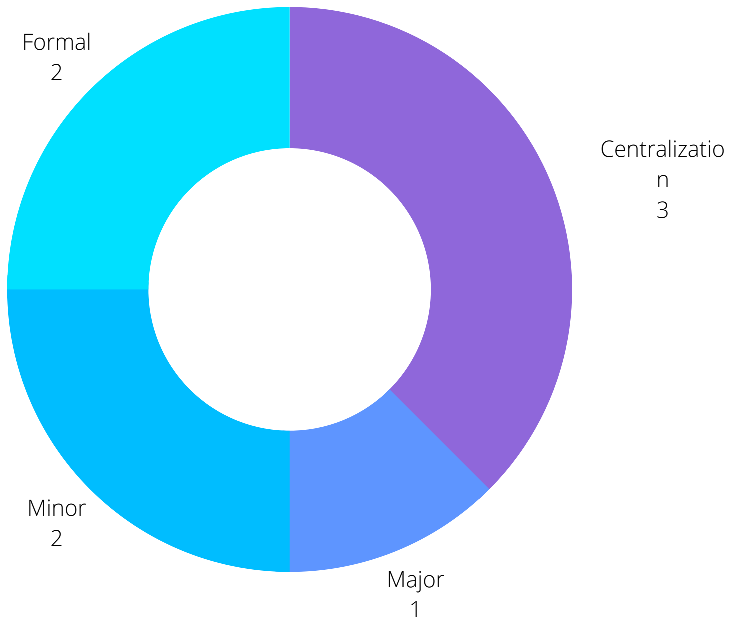
After our initial analysis, recommendations were made to the development team. All the issues found were completely resolved or adequately alleviated.

This document provides a full report of any isues found as well as the development team's response to them.

Overview

Project details	
Name	Cryptohots
Web	https://cryptohots.me/
Blockchain	Binance Smart Chain
Environment	EVM
Language	Solidity
Compiler version	0.8.13
Contract's Source	https://github.com/LuchoGuzman/CryptoHots.git
Contracts Audited	tokenhots.sol (commit 671bfb36538e5f42b3c280da7ad24e3974136c96) nft_classic_common.sol, nft_classic_infrecuent.sol, nft_classic_legenday.sol, nft_classic_mythic.sol, ft_classic_rare.sol, nft_season_rare.sol, nft_season_mythic.sol, nft_season_legenday.sol, nft_season_infrecuent.sol, nft_season_common.sol (commit 207bcf769ca8a4ad0c6f3d0b296b417f244c88b0)
Methodologies used	Automated check, manual check, line by line code review, tests deployment and live testing.

Findings summary



Formal: Comments on coding style and best practices issues. Mostly subjective.

Minor: Low-risk issues that cannot harm the contract's execution or expected behaviour.

Major: Medium-risk issues that can harm the contract or it's expected behaviour in a limited way.

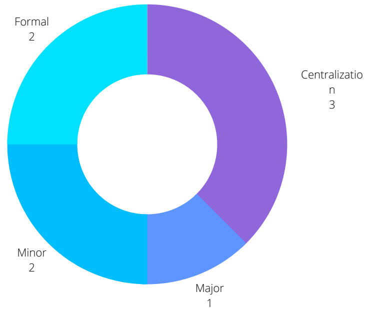
Critical: High-risk issues that can seriously harm the contract or compromise the ecosystem's security.

Centralization: Excess of centralized privilege can potentially represent a unfair playground for holders and investors.

Tokenhots.sol contract findings:

Severity Level	Found	Objected	Noted	Mitigated	Resolved
Formal	2	0	1	0	1
Minor	2	0	0	0	2
Major	1	0	0	0	1
Centralization	3	0	0	3	0
Critical	0	0	0	0	0

Findings details



Tokenhots.sol

Formal aspects FO

- FO1: Functions with unclear names that do not follow Solidity's best practices recommendations.

  Resolution: the team agreed to folllow Solidity's recommended naming conventions.

- FO2: Libraries safemath and safecast are imported but not actually used in the contract.

  Team's answer: We agreed they might be redundant but we feel they improve the contract's safety.

Minor Issues MI

- MI1: Functions requiring onlyOwner modifier could be declared external instead of public.

  Resolution: Function type declaration was updated.

- MI2: Upper fee limit not working as expected.

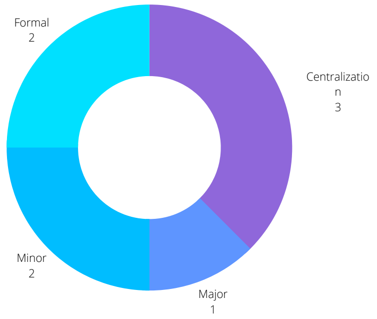
  Resolution: Code was updated and tested to work as intended.

Major Issues MA

- MA1: Potencially exploitable order dependence issue on transfer function.

  Resolution: Function have been rewritten to completely eliminate the risk.

Findings details



Tokenhots.sol

Centralization Issues CN

- Owner has several centralized privileges. In the event of foul play or compromise of the owner wallet, an attacker would be able to:
- CN1: Change fees, maximum transfer amount and maximum wallet size.
- CN2: Change the fee-receiving addresses.

Mitigation: limits were added to the centralized functions on findings CN2 and CN3 to eliminate the risk of honeypot.

- CN3: Pause the contract's execution.

Be sure you trust the project developers as they have considerable control. Ownership renounce would completely resolve any leftover centralization issues

Findings details

Erc721 contracts

Note: all the 11 erc721 contracts analysed share the same code, except for variables values. These findings apply to all of them.

Severity Level	Found	Objected	Noted	Mitigated	Resolved
Formal	0	0	0	0	0
Minor	0	0	0	0	0
Major	0	0	0	0	0
Centralization	1	0	0	1	0
Critical	0	0	0	0	0

Centralization Issues CN

- CN4: safeMint Function allows centralized minting of NFTs with no cost. Developers should inform their community about when and under which circumstances this function would be used.

Mitigation: Block explorer clearly shows whether the payable mint and the not payable safeMint is used in every case. So any attempt at exploiting the function would be public and visible in the blochain.

Final considerations

A total of 6 smart contracts were audited: An erc20 token (tokenhots.sol) and 10 non-fubgible erc721 NFT contracts (nft_season_mythic.sol, nft_season_legenday.sol, nft_season_infrecuent.sol, nft_season_common.sol, nft_classic_common.sol, nft_classic_infrecuent.sol, nft_classic_legenday.sol, nft_classic_mythic.sol nft_classic_rare.sol).

In the submitted ERC20 Token contract:

No critical or high risk issues were found.
One mayor, one formal and two minor issues were completely resolved in the updated version of the contract.
One formal issue was noted.
Three centralization issues were found. Two of them were adequately mitigated. The development team must add safeguard to the owner wallet, like a multisign system. Renouncing ownership would completely resolve any leftover centralization issues.

In the submitted ERC721 contracts:

One centralization issue was found. Transparency and good communication with the community must be ensured about when and why the team would use the ability to mint NFT without cost.

Version history

Version	Changelog
v0.1	• Initial review of contracts: tokenhots.sol, nft_classic_common.sol, nft_classic_infrecuent.sol, nft_classic_legenday.sol, nft_classic_mythic.sol, ft_classic_rare.sol • Findings sent to the development team.
v1.0	• (First published version) • Changes, mitigating actions and notes review. • Findings descriptions and posible mitigations updated.
v1.1	• Added review of contracts: nft_season_rare.sol, nft_season_mythic.sol, nft_season_legenday.sol, nft_season_infrecuent.sol, nft_season_common.sol
v1.2	• Review of updated contracts: nft_classic_common.sol, nft_classic_infrecuent.sol, nft_classic_legenday.sol, nft_classic_mythic.sol, ft_classic_rare.sol, nft_season_rare.sol, nft_season_mythic.sol, nft_season_legenday.sol, nft_season_infrecuent.sol, nft_season_common.sol, (commit 207bcf769ca8a4ad0c6f3d0b296b417f244c88b0) • Erc721 contracts findings and posible mitigations updated

Disclaimer

The information here provided is not, and must not be considered, endorsement, approval or disapproval of the audited project.
This audit report DOES NOT CONSTITUTE INVESTMENT ADVICE. It's scope is limited the technical and centralization aspects of the summited Smart Contracts.
Our team HAS NOT MADE ANY EVALUATION of the project's viability or economic design. Neither do we make any claims about the development team's ability, proficiency or well meaning.
This audit does not constitute any warranty of the absolute bug-free nature of the code or associated technologies used. Neither can it foresee any unwanted results of its interaction with third party solutions like daps, exchanges, in-game databases, or similar.
Crypto Alert NFT is in NO WAY RESPONSIBLE for any harm, malfunction or asset loss you might incour while interacting with the smart contracts here evaluated, nor is in any way liable for any detrimental results from your interaction with any aspects of the evaluated project.

April 2022

CRYPTO
ALERT
NFT

BLOCKCHAIN SECURITY ANALYSIS GROUP

https://t.me/cryptoalertnft (announcements)

https://t.me/scamalertnft (community)

https://scamalertnftgame.medium.com/ (web)