SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS OF:
FANCENTRIC

29th March 2022

TABLE OF CONTENTS

TABLE OF CONTENTS ---- 2
DISCLAIMER ---- 3
NOTE ---- 4
PROJECT SUMMARY ---- 5
Audit Result: ---- 6
SECURITY SCORE ---- 6
OVERVIEW ---- 7
Vulnerability Checklist ---- 9
Recommendations. ---- 12

DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest. Coinlens Team will take no payment for manipulating the results of this audit. The score and the result will stay on this project page information on our website https://coinlens.net. Coinlens Team does not guarantee that a project will not sell off team supply, or any other scam strategy (RUG or Honeypot etc.)

NOTE

Coinlens received an application for smart contract security audit of FANCENTRIC on March 28th March 2022, from the project team to discover if any vulnerability in the source code of the FANCETRIC project as well as any contract dependencies. Standard tests have been performed using Static Analysis and Manual Review techniques.

The auditing process focuses to the following considerations with collaboration of an expert team:

- Functionality test of the Smart Contract to determine if proper logic has been followed throughout the whole process.
- Manually detailed examination of the code line by line by experts.
- Live test by multiple clients using Testnet.
- Analyzing failure preparations to check how the Smart Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analyzing the security of the on-chain data.

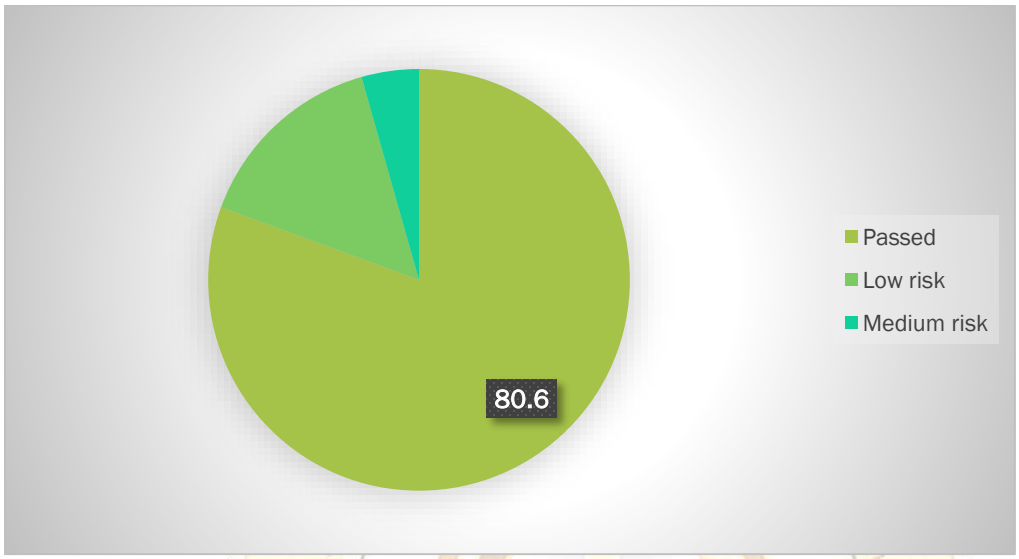
PROJECT SUMMARY

Project Name:	**FANCENTRIC TOKEN**
Website:	https://www.fancentric.co/
Twitter:	https://twitter.com/FanCentricNow
Telegram:	https://t.me/FanCentric
Platform:	Binance Smart Chain Token Type BEP20
Language:	Solidity
Contract address:	**0x82f5796c17df84C0898FdA0566aCF0e9459D9231**

Audit Result:

FANCENTRIC TOKEN has successfully **PASSED** the smart contract audit with **MEDIUM** level severity issues.

SECURITY SCORE
80.6



Audit Result:	**Passed**
Verified contract source	✓
No prior similar token contracts	✓
Ownership:	**Not renounced yet (Presale)**
KYC Verification:	**PINKSALE KYC at the date of this review**
Audit Date:	**March 29, 2022**
Audit Team:	**COINLENS**

OVERVIEW

This Audit Report mainly focuses on overall security of **FANCENTRIC TOKEN** Smart Contract.

CoinLens team scanned the contract and assessed overall system architecture and the smart contract codebase against vulnerabilities, exploitations, hacks, and back-doors to ensure its reliability and correctness.

Auditing Approach and Applied Methodologies

Coinlens team has performed rigorous test procedures of the project:

- ☺ Code design patterns analysis in which smart contract architecture is reviewed to ensure it is structured according to industry standards and safe use of third-party smart contracts and libraries.
- ☺ Line-by-line inspection of the Smart Contract to find any potential vulnerability like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.
- ☺ Unit testing Phase, we conducted custom unit tests written for each function in the contract to verify that each function works as expected.
- ☺ Simulated Test performed with our in-house developed tools to identify vulnerabilities and security flaws of the Smart Contract.

The focus of the audit was to verify that the Smart Contract System is secure, resilient, and working according to the specifications. The audit activities can be grouped in the following three categories:

Security

Identifying security related issues within each contract and the system of contract.

Code Correctness and Quality.

A full review of the contract source code. The primary focus included:

- ☺ Accuracy
- ☺ Readability
- ☺ Sections of code with high complexity
- ☺ Quantity and quality of test coverage

Risk Classification Vulnerabilities are classified in 3 main levels as below based on possible effect to the contract.
High level vulnerability Vulnerabilities on this level must be fixed immediately as they might lead to fund and data loss and open to manipulation. Any High-level finding will be highlighted with **RED** text.
Medium level vulnerability Vulnerabilities on this level also important to fix as they have potential risk of future exploit and manipulation. Any Medium-level finding will be highlighted with **ORANGE** text.
Low level vulnerability Vulnerabilities on this level are minor and may not affect the smart contract execution. Any Low-level finding will be highlighted with **BLUE** text

Vulnerability Checklist

No.	Description	Result
a)	Compiler warnings.	**Passed**
b)	Race conditions and reentrancy. Cross-function race Conditions.	**Passed**
c)	Possible delays in data delivery.	**Passed**
d)	Fallback function security.	**Passed**
e)	Cross-function race conditions.	**Passed**
f)	Oracle calls.	**Passed**
g)	DoS with Revert.	**Passed**
h)	Timestamp dependence.	**Passed**
i)	Integer Overflow and Underflow.	**Passed**
j)	Private user data leaks.	**Passed**
k)	Malicious Event log.	**Passed**
l)	Economy model.	**Passed**

Contract Analysis

Description	Status	Notes
No prior similar token contracts	✓	
Source does not contain a mint function	✗	The source code contains a mint function which could potentially allow new tokens to be created and dumped.
Ownership renounced or source does not contain an owner contract	✗	The contract contains ownership functionality and ownership is not renounced which may allow the creator or current owner to modify contract behavior (for example: disable selling, change fees, or mint new tokens). There can be legitimate reasons for not renouncing ownership, check with the project team for such information.
Verified contract source	✓	
Token is sellable (not a honeypot) at this time	✓	
Blacklist	✓	Addresses can be blacklisted. Blacklisted addresses cannot buy/sell!

Privileges of Ownership

Description	Status
Automatic LP is going to Owner wallet	✗
Blacklist	✓
Owner can transfer LP Token	✓
Owner can mint more coins	✓
Contract owner can renounce ownership	✓
Contract owner can transfer ownership	✓
Contract owner can exclude an address from transactions.	✓
The owner of this smart-contract can disable the trading	✓
Trade Disable	✓

Manual Audit:

For this section the code was tested/read line by line by our developers. Additionally, Remix IDE's JavaScript VM and Kovan networks used to test the contract functionality.

Automated Audit.

Remix Compiler Warnings

It throws warnings by Solidity's compiler. No issues found.

Recommendations.

"The Developers are supposed to fix the mint function to avoid misuse of minted tokens".

NOTE.

Owner can mint (create) new tokens up to 250,000,000 (current *supply is 7,224,874) and sell them in market. This can lead to liquidity drain.*

```
function mint(address _to, uint256 _amount) public onlyOwner {
require(fancentricSupply().add(_amount) <= cap, "FANC: cap
exceeded");
_mint(_to, _amount);
_moveDelegates(address(0), _delegates[_to], _amount);
```

Important Notice for Investors

As Coinlens team we are mainly auditing the contract code to find out how it will be functioning, and risks which are hidden in the code if any. There are many factors that must be taken into consideration before investing to a project, like:

- Ownership status
- Project team focus
- Marketing
- General market condition
- Liquidity
- Token holdings and roadmap etc.

Investors must always do their own research and manage their risk considering different factors which can affect the success of a project.