

# HEXNINJA AUDITS



Security Assessment

**FIFA Finance Token**

July 24, 2022

# Table of Contents

## **1 Audit Summary**

## **2 Project Overview**

### 2.1 Token Summary

### 2.2 Main Contract Assessed

## **3 Smart Contract Vulnerability Checks**

### 3.1 Mint Check

### 3.2 Fees Check

### 3.3 MaxTx Check

### 3.4 Pause Trade Check

## **4 Contract Ownership**

## **5 Liquidity Ownership**

## **6 Important Notes To The Users**

## **7 Social Media Check(Informational)**

## **8 Disclaimer**



# Audit Summary

This report has been prepared for FIFA Finance Token on the Binance Smart Chain network. 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       | 0x82b50CC6Dfbc0DAaC4816BD78b73632787B022E6                                                                                                                            |
| Name          | FIFA Finance                                                                                                                                                          |
| Token Tracker | FIFA Finance (FIFI)                                                                                                                                                   |
| Decimals      | 18                                                                                                                                                                    |
| Supply        | 1,000,000,000                                                                                                                                                         |
| Platform      | Binance Smart Chain                                                                                                                                                   |
| compiler      | v0.6.12+commit.27d51765                                                                                                                                               |
| Contract Name | FIFAFinance                                                                                                                                                           |
| Optimization  | Yes with 200 runs                                                                                                                                                     |
| LicenseType   | MIT                                                                                                                                                                   |
| Language      | Solidity                                                                                                                                                              |
| Codebase      | <a href="https://bscscan.com/address/0x82b50CC6Dfbc0DAaC4816BD78b73632787B022E6#code">https://bscscan.com/address/0x82b50CC6Dfbc0DAaC4816BD78b73632787B022E6#code</a> |
| Payment Tx    | 0x93cce45bdd2c80cd926735631a21d70a4fb03162d207ef07797fbb8545293efb                                                                                                    |



# Project Overview

## Risk Analysis Summary

| Parameter        | Result    |
|------------------|-----------|
| Buy Tax          | 9%        |
| Sale Tax         | 9%        |
| Is honeypot?     | Not Clean |
| Can edit tax?    | Yes       |
| Is anti whale?   | No        |
| Is blacklisted?  | No        |
| Is whitelisted?  | No        |
| Holders          | Not Clean |
| Security Score   | 50/100    |
| Auditor Score    | 90/100    |
| Confidence Level | Low       |

The following quick summary has been added to the project overview, however there are more details about the audit and their results please read every details.



## Main Contract Assessed Contract Name

| Name         | Contract                                   | Live |
|--------------|--------------------------------------------|------|
| FIFA Finance | 0x82b50CC6Dfbc0DAaC4816BD78b73632787B022E6 | Yes  |

## TestNet Contract Assessed Contract Name

| Name         | Contract                                   | Live |
|--------------|--------------------------------------------|------|
| FIFA Finance | 0x0e0A1ccaab40969ad1aCEA0525A28f9cE6A672fB | Yes  |

## Solidity Code Provided

| SolID | File Sha-1                               | FileName |
|-------|------------------------------------------|----------|
| FIF1  | 034240bc3e5194471cee0b93c1dc25f7fd438275 | fifa.sol |
| FIF1  |                                          |          |
| FIF1  |                                          |          |



# 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 |



# Mint Check

The Project Owners of FIFA Finance does not have a mint function in the contract, owner cannot mint tokens after initial deploy

..

The Project has a Total Supply of 1,000,000,000 and cannot mint any more than the Max Supply.

.

## Mint Notes:

**Auditor Notes:** A Mint Function was not found during the code review

**Project Owner Notes:**



**Owner can't mint new coins**



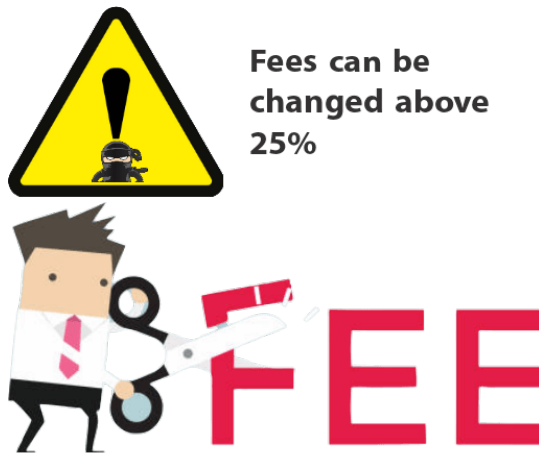
# Fees Check

The Project Owners of FIFA Finance has the ability to set fees to 50%  
We Recommend the team to create a new contract with fees restrictions to avoid any problems, as alternative the team can use multi signature wallet to ensure the project is safe from a potential fee increase.

## Tax Fee Notes:

**Auditor Notes:** Contract currently have 9% tax and can be increased, if the customer use the setttee function it will break the contract. there is a log that revert to 10% and this logic does not work as intended

**Project Owner Notes:** .



# MaxTx Check

The Project Owners of FIFA Finance does not has the ability to set max tx amount

The Team allow any investors to swap, transfer or sale their total amount if needed.

**MaxTX Notes:**

**Auditor Notes:** No max tx found.

**Project Owner Notes:**

**Project Has No MaxTX**



# Pause Trade Check

The Project Owners of FIFA Finance Owner can pause trading but he can't move tokens  
(Owner can't pause trading)

The Team has done a great job to avoid stop trading, and investors has the ability to trade  
at any given time without any problems

**Pause Trade Notes:**

**Auditor Notes: Not found.**

**Project Owner Notes:**



**Owner can't pause trading**



# Contract Ownership

The contract ownership of FIFA Finance 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 `0x9562381A8C497cDd3dae2126DD71e911D2Ad9730` which can be viewed from:  
[HERE](#)

The owner wallet has the power to call the functions displayed on the privileged 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.

# Liquidity Ownership

The token does not have liquidity at the moment of the audit, block 19812659



# KYC Information

The Project Owners of FIFA Finance is not KYC. .

The owner wallet has the power to call the functions displayed on the privileged 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.

**KYC Information Notes:**

**Auditor Notes:** Asked project owner about KYC or Doxxed

**Project Owner Notes:**



# Mythx Security Summary Checks

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



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



| ID      | Severity | Name                                                     | File     | location  |
|---------|----------|----------------------------------------------------------|----------|-----------|
| SWC-129 | Pass     | Typographical Error.                                     | fifa.sol | L: 0 C: 0 |
| SWC-130 | Pass     | Right-To-Left-Override control character (U+202E).       | fifa.sol | L: 0 C: 0 |
| SWC-131 | Pass     | Presence of unused variables.                            | fifa.sol | L: 0 C: 0 |
| SWC-132 | Pass     | Unexpected Ether balance.                                | fifa.sol | L: 0 C: 0 |
| SWC-133 | Pass     | Hash Collisions with Multiple Variable Length Arguments. | fifa.sol | L: 0 C: 0 |
| SWC-134 | Pass     | Message call with hardcoded gas amount.                  | fifa.sol | L: 0 C: 0 |
| SWC-135 | Pass     | Code With No Effects (Irrelevant/Dead Code).             | fifa.sol | L: 0 C: 0 |
| SWC-136 | Pass     | Unencrypted Private Data On-Chain.                       | fifa.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-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-120 – Weak Sources of Randomness from Chain Attributes

## CWE-330: Use of Insufficiently Random Values

### Description:

Solidity allows for ambiguous naming of state variables when inheritance is used. Contract A with a variable x could inherit contract B that also has a state variable x defined. This would result in two separate versions of x, one of them being accessed from contract A and the other one from contract B. In more complex contract systems this condition could go unnoticed and subsequently lead to security issues.

Shadowing state variables can also occur within a single contract when there are multiple definitions on the contract and function level.

### Remediation:

Using commitment scheme, e.g. RANDAO. Using external sources of randomness via oracles, e.g. Oraclize. Note that this approach requires trusting in oracle, thus it may be reasonable to use multiple oracles. Using Bitcoin block hashes, as they are more expensive to mine.

## References:

How can I securely generate a random number in my smart contract?)

When can BLOCKHASH be safely used for a random number? When would it be unsafe?

The Run smart contract.

### SWC Information Notes:

**Auditor Notes:** No Vulnerabilities where found during the security scan.

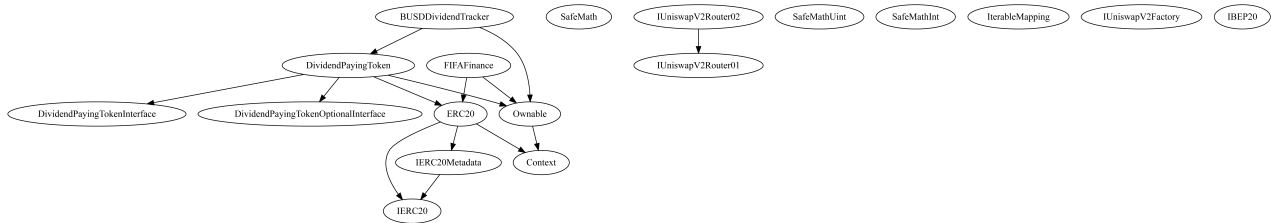
**Project Owner Notes:**



# Call Graph and Inheritance

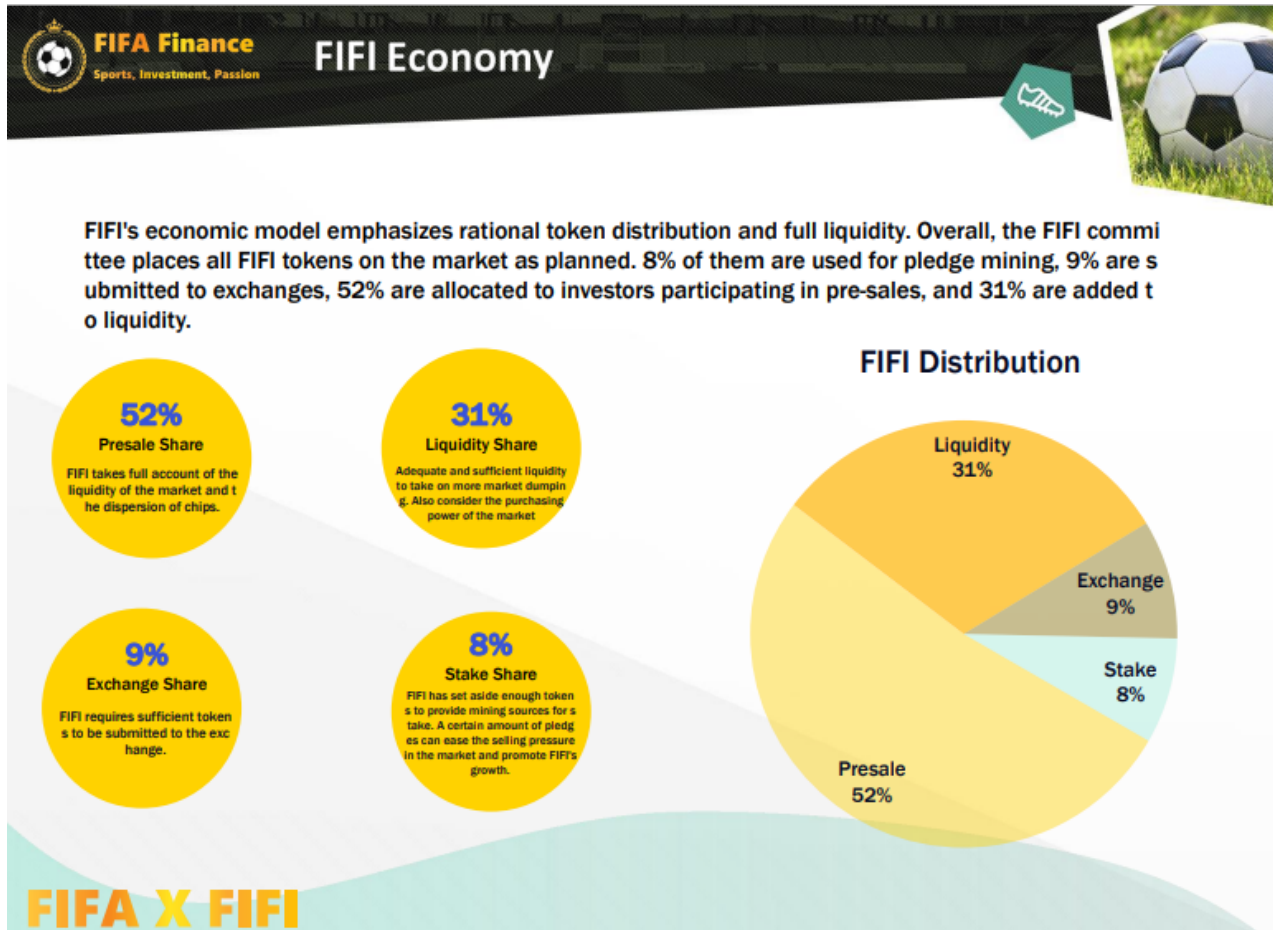
The contract for FIFA Finance has the following call graph structure

The Project has a Total Supply of 1,000,000,000 and has the following inheritance



# Tokenomics

The contract for FIFA Finance has the following tokenomics.



## Privileged Functions (onlyOwner)

| Function Name               | Parameters                                                | Visibility |
|-----------------------------|-----------------------------------------------------------|------------|
| renounceOwnership           | none                                                      | public     |
| transferOwnership           | address newOwner                                          | public     |
| setFee                      | _BUSDRewardsFee (uint256),<br>_BUSDRewardsShare (uint256) | public     |
| updateDividendTracker       | none                                                      | external   |
| rescueToken                 | _tokenAddress (address),<br>_amount (uint256)             | external   |
| rescueBNB                   |                                                           | public     |
| updateUniswapV2Router       |                                                           | public     |
| setSwapAndLiquifyEnabled    | _enabled (bool)                                           | public     |
| setDevWallet                | Wallet (address)                                          | external   |
| setBurnWallet               | Wallet (address)                                          | external   |
| setPresaleContract          | Wallet (address)                                          | external   |
| setFoudWalletAddress        | Wallet (address)                                          | external   |
| setAutomatedMarketMakerPair |                                                           | external   |
| includeInFee                | account (address)                                         | external   |
| excludeFromFee              | account (address)                                         | external   |



| Function Name          | Parameters | Visibility |
|------------------------|------------|------------|
| updateGasForProcessing |            | external   |
| updateClaimWait        |            | external   |
| excludeFromDividends   |            | external   |
| setSnipeBlocks         |            | external   |



## Auditors Final Verdict

### Important Notes To The Users:

- Owner can charge fees up to 10%, however, there is a problem with this function
- Owner can't set max tx amount.
- Owner can't pause trading.
- No high-risk Exploits/Vulnerabilities Were Found in the Source Code.
- Reviewed Whitepaper and seems ok, however, there are a few minor discrepancies.

## Audit Passed



# Social Media Checks

| Social Media | URL                                                                           | Result |
|--------------|-------------------------------------------------------------------------------|--------|
| Twitter      | <a href="https://twitter.com/FIFAFinance">https://twitter.com/FIFAFinance</a> | Pass   |
| Instagram    |                                                                               | Fail   |
| Website      | <a href="https://www.fifa.finance/">https://www.fifa.finance/</a>             | Pass   |
| Telegram     | <a href="https://t.me/FIFAFinanceGlobal">https://t.me/FIFAFinanceGlobal</a>   | 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:** No other social media



## FIFI-01 | Potential Sandwich Attacks.

| Category | Severity                                                                                 | Location         | Status                                                                                      |
|----------|------------------------------------------------------------------------------------------|------------------|---------------------------------------------------------------------------------------------|
| Security |  Medium | fifa.sol: 1852,0 |  Pending |

### Description:

A sandwich attack might happen when an attacker observes a transaction swapping tokens or adding liquidity without setting restrictions on slippage or minimum output amount. The attacker can manipulate the exchange rate by frontrunning (before the transaction being attacked) a transaction to purchase one of the assets and make profits by back running (after the transaction being attacked) a transaction to sell the asset. The following functions are called without setting restrictions on slippage or minimum output amount, so transactions triggering these functions are vulnerable to sandwich attacks, especially when the input amount is large:

- swapExactTokensForETHSupportingFeeOnTransferTokens()
- addLiquidityETH()

### Remediation:

We recommend setting reasonable minimum output amounts, instead of 0, based on token prices when calling the aforementioned functions.

### References:

What Are Sandwich Attacks in DeFi – and How Can You Avoid Them?.



## FIFI-02 | Function Visibility Optimization.

| Category         | Severity                                                                                        | Location      | Status                                                                                      |
|------------------|-------------------------------------------------------------------------------------------------|---------------|---------------------------------------------------------------------------------------------|
| Gas Optimization |  Informational | fifa.sol: 0,0 |  Pending |

### Description:

The following functions are declared as public and are not invoked in any of the contracts contained within the projects scope:

| Function Name     | Parameters                                                | Visibility |
|-------------------|-----------------------------------------------------------|------------|
| renounceOwnership | none                                                      | public     |
| transferOwnership | address newOwner                                          | public     |
| setFee            | _BUSDRewardsFee (uint256),<br>_BUSDRewardsShare (uint256) | public     |
| name              |                                                           | public     |
| symbol            |                                                           | public     |
| decimals          |                                                           | public     |

The functions that are never called internally within the contract should have external visibility

### Remediation:

We advise that the functions' visibility specifiers are set to external and the array-based arguments change their data location from memory to calldata , optimizing the gas cost of the function.

### References:

external vs public best practices.



## FIFI-03 | Lack of Input Validation.

| Category      | Severity                                                                                | Location      | Status                                                                                      |
|---------------|-----------------------------------------------------------------------------------------|---------------|---------------------------------------------------------------------------------------------|
| Volatile Code |  Minor | fifa.sol: 0,0 |  Pending |

### Description:

The given input is missing the check for the non-zero address.

### Remediation:

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
...  
require(receiver != address(0), "Receiver is the zero address");  
...
```



## FIFI-04 | Centralized Risk In addLiquidity.

| Category     | Severity | Location      | Status    |
|--------------|----------|---------------|-----------|
| Coding Style | 🟡 Major  | fifa.sol: 0,0 | 📋 Pending |

### Description:

```
uniswapV2Router.addLiquidityETH{value: ethAmount}(address(this), tokenAmount, 0, 0, owner(), block.timestamp);
```

The addLiquidity function calls the uniswapV2Router.addLiquidityETH function with the to address specified as owner() for acquiring the generated LP tokens from the FIFI-WBNB pool.

As a result, over time the \_owner address will accumulate a significant portion of LP tokens. If the \_owner is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

### Remediation:

We advise the to address of the uniswapV2Router.addLiquidityETH function call to be replaced by the contract itself, i.e. address(this) , and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the \_owner account is compromised. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or via smart-contract based accounts with enhanced security practices, f.e. Multisignature wallets.

Indicatively, here are some feasible solutions that would also mitigate the potential risk:

- Time-lock with reasonable latency, i.e. 48 hours, for awareness on privileged operations;

- Assignment of privileged roles to multi-signature wallets to prevent single point of failure due to the private key;

- Introduction of a DAO / governance / voting module to increase transparency and user involvement");



## FIFI-05 | Missing Event Emission.

| Category      | Severity                                                                                | Location      | Status                                                                                      |
|---------------|-----------------------------------------------------------------------------------------|---------------|---------------------------------------------------------------------------------------------|
| Volatile Code |  Major | fifa.sol: 0,0 |  Pending |

### Description:

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes. The linked code does not create an event for the transfer.

### Remediation:

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.  
");



# Disclaimer

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

