



# SMART CONTRACT AUDIT



interfinetwork



hello@interfi.network



<https://interfi.network>

PREPARED FOR

**NOVOOS TOKEN**



# INTRODUCTION

Auditing Firm	InterFi Network
Client Firm	Novoos
Methodology	Automated Analysis, Manual Code Review
Language	Solidity
Token Proxy	0xD655981fB2F15498e3279EC34CAe2baC6c6744CD
Token Implementation	0x4c105Ffe6319D0E65BA805C9934Bde24518d616a
Blockchain	Binance Smart Chain
Centralization	Active ownership
Commit	02b83590f1726fded935387b7d5d2b442acbec63
Website	<a href="https://novoos.net/en/">https://novoos.net/en/</a>
Telegram	<a href="https://t.me/novoosecosystem/">https://t.me/novoosecosystem/</a>   <a href="https://t.me/Novoosannouncements/">https://t.me/Novoosannouncements/</a>
Twitter	<a href="https://twitter.com/Novotoken/">https://twitter.com/Novotoken/</a>
Socials	<a href="https://novoos.net/en/novoos-official-links/">https://novoos.net/en/novoos-official-links/</a>
Report Date	March 11, 2023

 Verify the authenticity of this report on our website: <https://www.github.com/interfinetwork>



## EXECUTIVE SUMMARY

InterFi has performed the automated and manual analysis of solidity codes. Solidity codes were reviewed for common contract vulnerabilities and centralized exploits. Here's a quick audit summary:

Status	Critical <span style="color: red;">●</span>	Major <span style="color: orange;">●</span>	Medium <span style="color: yellow;">●</span>	Minor <span style="color: green;">●</span>	Unknown <span style="color: brown;">●</span>
Open	0	0	0	3	0
Acknowledged	0	0	0	0	0
Resolved	1	1	1	3	1
Noteworthy Privileges	Review PAGE 18 for centralization related privileges				

**i** Please note that smart contracts deployed on blockchains aren't resistant to exploits, vulnerabilities and/or hacks. Blockchain and cryptography assets utilize new and emerging technologies. These technologies present a high level of ongoing risks. For a detailed understanding of risk severity, source code vulnerability, and audit limitations, kindly review the audit report thoroughly.

**i** Please note that centralization privileges regardless of their inherited risk status - constitute an elevated impact on smart contract safety and security.



# TABLE OF CONTENTS

TABLE OF CONTENTS .....	4
SCOPE OF WORK .....	5
AUDIT METHODOLOGY .....	6
RISK CATEGORIES.....	8
CENTRALIZED PRIVILEGES.....	9
AUTOMATED ANALYSIS .....	10
INHERITANCE GRAPH.....	17
MANUAL REVIEW .....	18
DISCLAIMERS.....	32
ABOUT INTERFI NETWORK.....	35

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL



## SCOPE OF WORK

InterFi was consulted by Novoos to conduct the smart contract audit of their solidity source codes. The audit scope of work is strictly limited to mentioned solidity file(s) only:

- Novoos.sol

 If source codes are not deployed on the main net, they can be modified or altered before main-net deployment. Verify the contract's deployment status below:

Public Contract Link	
<a href="https://bscscan.com/address/0x4c105ffe6319d0e65ba805c9934bde24518d616a#code">https://bscscan.com/address/0x4c105ffe6319d0e65ba805c9934bde24518d616a#code</a>	
Contract Name	Novoos
Compiler Version	0.8.17
License	MIT



# AUDIT METHODOLOGY

Smart contract audits are conducted using a set of standards and procedures. Mutual collaboration is essential to performing an effective smart contract audit. Here's a brief overview of InterFi's auditing process and methodology:

## CONNECT

- The onboarding team gathers source codes, and specifications to make sure we understand the size, and scope of the smart contract audit.

## AUDIT

- Automated analysis is performed to identify common contract vulnerabilities. We may use the following third-party frameworks and dependencies to perform the automated analysis:
  - Remix IDE Developer Tool
  - Open Zeppelin Code Analyzer
  - SWC Vulnerabilities Registry
  - DEX Dependencies, e.g., Pancakeswap, Uniswap
- Simulations are performed to identify centralized exploits causing contract and/or trade locks.
- A manual line-by-line analysis is performed to identify contract issues and centralized privileges.

We may inspect below mentioned common contract vulnerabilities, and centralized exploits:

Centralized Exploits	<ul style="list-style-type: none"><li>○ Token Supply Manipulation</li><li>○ Access Control and Authorization</li><li>○ Assets Manipulation</li><li>○ Ownership Control</li><li>○ Liquidity Access</li><li>○ Stop and Pause Trading</li><li>○ Ownable Library Verification</li></ul>
----------------------	---



## Common Contract Vulnerabilities

- Integer Overflow
- Lack of Arbitrary limits
- Incorrect Inheritance Order
- Typographical Errors
- Requirement Violation
- Gas Optimization
- Coding Style Violations
- Re-entrancy
- Third-Party Dependencies
- Potential Sandwich Attacks
- Irrelevant Codes
- Divide before multiply
- Conformance to Solidity Naming Guides
- Compiler Specific Warnings
- Language Specific Warnings

**REPORT**

- The auditing team provides a preliminary report specifying all the checks which have been performed and the findings thereof.
- The client's development team reviews the report and makes amendments to solidity codes.
- The auditing team provides the final comprehensive report with open and unresolved issues.

**PUBLISH**

- The client may use the audit report internally or disclose it publicly.

 It is important to note that there is no pass or fail in the audit, it is recommended to view the audit as an unbiased assessment of the safety of solidity codes.



## RISK CATEGORIES

Smart contracts are generally designed to hold, approve, and transfer tokens. This makes them very tempting attack targets. A successful external attack may allow the external attacker to directly exploit. A successful centralization-related exploit may allow the privileged role to directly exploit. All risks which are identified in the audit report are categorized here for the reader to review:

Risk Type	Definition
Critical 	These risks could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.
Major 	These risks are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to high-risk severity.
Medium 	These risks should be fixed, as they carry an inherent risk of future exploits, and hacks which may or may not impact the smart contract execution. Low-risk re-entrancy-related vulnerabilities should be fixed to deter exploits.
Minor 	These risks do not pose a considerable risk to the contract or those who interact with it. They are code-style violations and deviations from standard practices. They should be highlighted and fixed nonetheless.
Unknown 	These risks pose uncertain severity to the contract or those who interact with it. They should be fixed immediately to mitigate the risk uncertainty.

All statuses which are identified in the audit report are categorized here for the reader to review:

Status Type	Definition
Open	Risks are open.
Acknowledged	Risks are acknowledged, but not fixed.
Resolved	Risks are acknowledged and fixed.





## CENTRALIZED PRIVILEGES

Centralization risk is the most common cause of cryptography asset loss. When a smart contract has a privileged role, the risk related to centralization is elevated.

There are some well-intended reasons have privileged roles, such as:

- Privileged roles can be granted the power to pause() the contract in case of an external attack.
- Privileged roles can use functions like, include(), and exclude() to add or remove wallets from fees, swap checks, and transaction limits. This is useful to run a presale and to list on an exchange.

Authorizing privileged roles to externally-owned-account (EOA) is dangerous. Lately, centralization-related losses are increasing in frequency and magnitude.

- The client can lower centralization-related risks by implementing below mentioned practices:
- Privileged role's private key must be carefully secured to avoid any potential hack.
- Privileged role should be shared by multi-signature (multi-sig) wallets.
- Authorized privilege can be locked in a contract, user voting, or community DAO can be introduced to unlock the privilege.
- Renouncing the contract ownership, and privileged roles.
- Remove functions with elevated centralization risk.

 Understand the project's initial asset distribution. Assets in the liquidity pair should be locked. Assets outside the liquidity pair should be locked with a release schedule.



## AUTOMATED ANALYSIS

Symbol	Definition
	Function modifies state
	Function is payable
	Function is internal
	Function is private
	Function is important

```

| **Initializable** | Implementation | |||
| ^ | _disableInitializers | Internal  |  | |
| ^ | _getInitializedVersion | Internal  | | |
| ^ | _isInitializing | Internal  | | |
|||||
| **AddressUpgradeable** | Library | |||
| ^ | isContract | Internal  | | |
| ^ | sendValue | Internal  |  | |
| ^ | functionCall | Internal  |  | |
| ^ | functionCall | Internal  |  | |
| ^ | functionCallWithValue | Internal  |  | |
| ^ | functionCallWithValue | Internal  |  | |
| ^ | functionStaticCall | Internal  | | |
| ^ | functionStaticCall | Internal  | | |
| ^ | verifyCallResultFromTarget | Internal  | | |
| ^ | verifyCallResult | Internal  | | |
| ^ | _revert | Private  | | |
|||||

```

INTERFI  
CONFIDENTIAL



```

| **Context** | Implementation |   | | |
|  L | _msgSender | Internal 🔒 |   |
|  L | _msgData | Internal 🔒 |   |
|||||
| **Ownable** | Implementation | Context, Initializable |
|  L | initialize | Public ! | 🚫 | initializer |
|  L | owner | Public ! | |NO! |
|  L | renounceOwnership | Public ! | 🚫 | onlyOwner |
|  L | transferOwnership | Public ! | 🚫 | onlyOwner |
|||||
| **IERC20** | Interface |   |
|  L | totalSupply | External ! | |NO! |
|  L | balanceOf | External ! | |NO! |
|  L | transfer | External ! | 🚫 |NO! |
|  L | allowance | External ! | |NO! |
|  L | approve | External ! | 🚫 |NO! |
|  L | transferFrom | External ! | 🚫 |NO! |
|||||
| **ERC20** | Implementation | Context, IERC20, Initializable |
|  L | initialize | Public ! | 🚫 | initializer |
|  L | name | Public ! | |NO! |
|  L | symbol | Public ! | |NO! |
|  L | decimals | Public ! | |NO! |
|  L | totalSupply | Public ! | |NO! |
|  L | balanceOf | Public ! | |NO! |
|  L | transfer | Public ! | 🚫 |NO! |
|  L | allowance | Public ! | |NO! |
|  L | approve | Public ! | 🚫 |NO! |

```

INTERFI  
CONFIDENTIAL

```

|  L | transferFrom | Public ! |  | NO ! |
|  L | increaseAllowance | Public ! |  | NO ! |
|  L | decreaseAllowance | Public ! |  | NO ! |
|  L | _transfer | Internal  |  |  |
|  L | _mint | Internal  |  |  |
|  L | _burn | Internal  |  |  |
|  L | _approve | Internal  |  |  |
|  L | _setupDecimals | Internal  |  |  |
|  L | _beforeTokenTransfer | Internal  |  |  |
|||||

```

```

| **IUniswapV2Factory** | Interface | |||
|  L | feeTo | External ! |  | NO ! |
|  L | feeToSetter | External ! |  | NO ! |
|  L | getPair | External ! |  | NO ! |
|  L | allPairs | External ! |  | NO ! |
|  L | allPairsLength | External ! |  | NO ! |
|  L | createPair | External ! |  | NO ! |
|  L | setFeeTo | External ! |  | NO ! |
|  L | setFeeToSetter | External ! |  | NO ! |
|||||

```

```

| **IUniswapV2Router01** | Interface | |||
|  L | factory | External ! |  | NO ! |
|  L | WETH | External ! |  | NO ! |
|  L | addLiquidity | External ! |  | NO ! |
|  L | addLiquidityETH | External ! |  | NO ! |
|  L | removeLiquidity | External ! |  | NO ! |
|  L | removeLiquidityETH | External ! |  | NO ! |
|  L | removeLiquidityWithPermit | External ! |  | NO ! |

```

TERFI  
CONFIDENTIAL

INTERFI  
CONFIDENTIAL



```

|  L | removeLiquidityETHWithPermit | External ! | ● | NO ! |
|  L | swapExactTokensForTokens | External ! | ● | NO ! |
|  L | swapTokensForExactTokens | External ! | ● | NO ! |
|  L | swapExactETHForTokens | External ! | 🚫 | NO ! |
|  L | swapTokensForExactETH | External ! | ● | NO ! |
|  L | swapExactTokensForETH | External ! | ● | NO ! |
|  L | swapETHForExactTokens | External ! | 🚫 | NO ! |
|  L | quote | External ! | | NO ! |
|  L | getAmountOut | External ! | | NO ! |
|  L | getAmountIn | External ! | | NO ! |
|  L | getAmountsOut | External ! | | NO ! |
|  L | getAmountsIn | External ! | | NO ! |
|||||

```

```

| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||
|  L | removeLiquidityETHSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|  L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|  L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|  L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | 🚫 | NO ! |
|  L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | ● | NO ! |
|||||


```


```


| **SafeMath** | Library | |||
|  L | tryAdd | Internal 🚫 | | |
|  L | trySub | Internal 🚫 | | |
|  L | tryMul | Internal 🚫 | | |
|  L | tryDiv | Internal 🚫 | | |
|  L | tryMod | Internal 🚫 | | |
|  L | add | Internal 🚫 | | |
|  L | sub | Internal 🚫 | | |


```


INTERFI  
CONFIDENTIAL


| <sup>L</sup> | mul | Internal  | | |

| <sup>L</sup> | div | Internal  | | |

| <sup>L</sup> | mod | Internal  | | |


| <sup>L</sup> | sub | Internal  | | |


| <sup>L</sup> | div | Internal  | | |


| <sup>L</sup> | mod | Internal  | | |


|||||


| **\*\*SafeMathInt\*\*** | Library | |||

| <sup>L</sup> | mul | Internal  | | |

| <sup>L</sup> | div | Internal  | | |

| <sup>L</sup> | sub | Internal  | | |

| <sup>L</sup> | add | Internal  | | |

| <sup>L</sup> | toUint256Safe | Internal  | | |


|||||

| **\*\*SafeMathUint\*\*** | Library | |||


| <sup>L</sup> | toInt256Safe | Internal  | | |


|||||

| **\*\*IDividendDistributor\*\*** | Interface | |||

| <sup>L</sup> | setDistributionCriteria | External ! |  | NO ! |


| <sup>L</sup> | setShare | External ! |  | NO ! |

| <sup>L</sup> | deposit | External ! |  | NO ! |

| <sup>L</sup> | process | External ! |  | NO ! |

|||||

| **\*\*IWETH\*\*** | Interface | |||

| <sup>L</sup> | deposit | External ! |  | NO ! |

| <sup>L</sup> | transfer | External ! |  | NO ! |

|||||

| **\*\*IJackpot\*\*** | Interface | |||

| <sup>L</sup> | userBuy | External ! |  | NO ! |

INTERFI  
CONFIDENTIAL

INTERFI  
CONFIDENTIAL



```

| L | deposit | External ! | 🚫 | NO ! |
|||||
| **Novoos** | Implementation | ERC20, Ownable |||
| L | initialize | Public ! | 🚫 | initializer |
| L | <Receive Ether> | External ! | 🚫 | NO ! |
| L | <Fallback> | External ! | 🚫 | NO ! |
| L | setWalletBalance | External ! | 🚫 | onlyOwner |
| L | setMaxBuyTransaction | External ! | 🚫 | onlyOwner |
| L | setMaxSellTransaction | External ! | 🚫 | onlyOwner |
| L | setAllowanceFundAddress | External ! | 🚫 | onlyOwner |
| L | setBankAddress | External ! | 🚫 | onlyOwner |
| L | setMarianatrenchAddress | External ! | 🚫 | onlyOwner |
| L | setJackpotAddress | External ! | 🚫 | onlyOwner |
| L | setInsuranceAndRewardFundAddress | External ! | 🚫 | onlyOwner |
| L | setLpFeeBuy | External ! | 🚫 | onlyOwner |
| L | setAllowanceFundFeeBuy | External ! | 🚫 | onlyOwner |
| L | setBankFundFeeBuy | External ! | 🚫 | onlyOwner |
| L | setMarianatrenchFeeBuy | External ! | 🚫 | onlyOwner |
| L | setBusdDividendRewardsFeeBuy | External ! | 🚫 | onlyOwner |
| L | setJackpotFeeBuy | External ! | 🚫 | onlyOwner |
| L | setInsuranceAndRewardFundFeeBuy | External ! | 🚫 | onlyOwner |
| L | setLpFeeSell | External ! | 🚫 | onlyOwner |
| L | setAllowanceFundFeeSell | External ! | 🚫 | onlyOwner |
| L | setBankFundFeeSell | External ! | 🚫 | onlyOwner |
| L | setMarianatrenchFeeSell | External ! | 🚫 | onlyOwner |
| L | setBusdDividendRewardsFeeSell | External ! | 🚫 | onlyOwner |
| L | setJackpotFeeSell | External ! | 🚫 | onlyOwner |
| L | setInsuranceAndRewardFundFeeSell | External ! | 🚫 | onlyOwner |

```

INTERFI  
CONFIDENTIAL

	└		setBusdRewardsStatus		External	!		🔴		onlyOwner	
	└		setJackpotStatus		External	!		🔴		onlyOwner	
	└		setSwapAndLiquifyEnabled		External	!		🔴		onlyOwner	
	└		updateUniswapV2Router		External	!		🔴		onlyOwner	
	└		excludeFromFees		Public	!		🔴		onlyOwner	
	└		setAutomatedMarketMakerPair		Public	!		🔴		onlyOwner	
	└		_setAutomatedMarketMakerPair		Private	🔒		🔴		onlyOwner	
	└		updateGasForProcessing		External	!		🔴		onlyOwner	
	└		setIsDividendExempt		External	!		🔴		onlyOwner	
	└		setDividendDistributor		External	!		🔴		onlyOwner	
	└		getIsExcludedFromFees		Public	!		NO !			
	└		setJackpotLimit		External	!		🔴		onlyOwner	
	└		_transfer		Internal	🔒		🔴			
	└		_getLiquidityImpact		Internal	🔒					
	└		_deactivateTripSwitch		Private	🔒		🔴			
	└		_getPriceChange		Internal	🔒					
	└		accuTaxSystem		Internal	🔒		🔴			
	└		depositAmountToBusdReward		Private	🔒		🔴			
	└		depositAmountToJackpot		Private	🔒		🔴			
	└		swapAndLiquify		Private	🔒		🔴			
	└		addLiquidity		Private	🔒		🔴			
	└		swapTokensForBNB		Private	🔒		🔴			
	└		swapTokensForDividendToken		Private	🔒		🔴			
	└		setTripSwitchDuration		External	!		🔴		onlyOwner	
	└		setTripSwitchPeriod		External	!		🔴		onlyOwner	
	└		setTripSwitchTriggerTarget		External	!		🔴		onlyOwner	

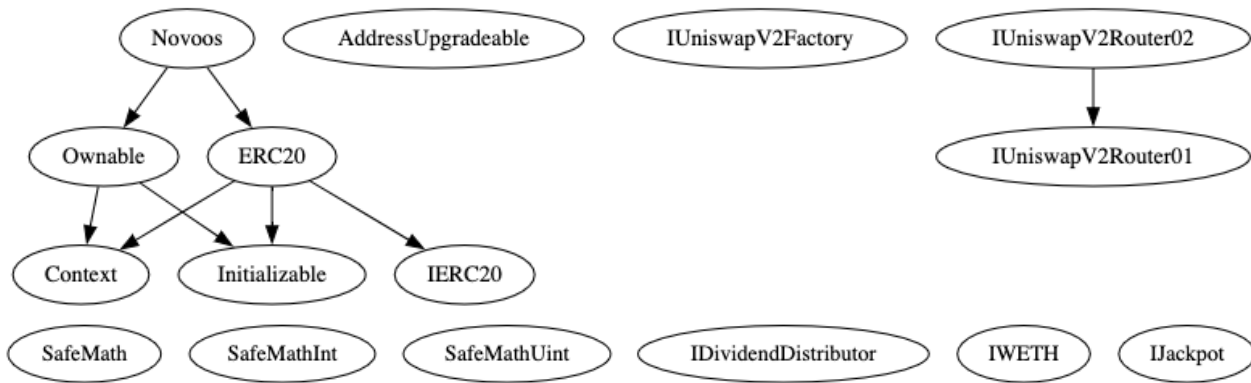
INTERFI  
CONFIDENTIAL

INTERFI  
CONFIDENTIAL





## INHERITANCE GRAPH



INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT



## MANUAL REVIEW

Identifier	Definition	Severity
CEN-01	Centralized privileges	Major 🟡

Important only owner centralized privileges are listed below:

```

transferOwnership()
setWalletBalance()
setMaxBuyTransaction()
setMaxSellTransaction()
setLpFeeBuy()
setAllowanceFundFeeBuy()
setBankFundFeeBuy()
setMarianatrenchFeeBuy()
setBusdDividendRewardsFeeBuy()
setJackpotFeeBuy()
setInsuranceAndRewardFundFeeBuy()
setLpFeeSell()
setAllowanceFundFeeSell()
setBankFundFeeSell()
setMarianatrenchFeeSell()
setBusdDividendRewardsFeeSell()
setJackpotFeeSell()
setInsuranceAndRewardFundFeeSell()
setBusdRewardsStatus()
setJackpotStatus()
setSwapAndLiquifyEnabled()
updateUniswapV2Router()
setAutomatedMarketMakerPair()
updateGasForProcessing()
setDividendDistributor()
setJackpotLimit()
setTripSwitchDuration()
setTripSwitchPeriod()
setTripSwitchTriggerTarget()

```

INTERFI  
CONFIDENTIAL



## RECOMMENDATION

Deployer, contract owner, and privileged roles' private keys should be secured carefully. Please refer to PAGE-09 CENTRALIZED PRIVILEGES for a detailed understanding.

## RESOLUTION

According to Novoos, some centralization is necessary and it is not available for public to trigger.

"These will assist to sustain the token as well various aspects of campaigns planned for the rewards, jackpot etc. for example our taxes currently are 10% Buy & 12% Sell. After launch, there is a campaign to reduce these and market it in order to inform individuals to "buy" as the buy taxes are now lower.

Our NovoPad NFT collection where it provides tax relief as one incentive is another aspect.

The privileges are set to the owner's address, it is public that we have the access and not a secret and neither hidden from the community.

We have a few pages on the main contact and much more information that it is an upgradeable and proxy contact:

- o <https://docs.novoos.net/usdnovo-ameliorative-contract/the-nac>
- o [https://docs.google.com/presentation/d/e/2PACX-1vRiU7zISF3MzvpwF3S0lvGxis6mvi70IfBgU8dWByxy\\_RSJ4gHNo5WpgVOZYbcoXXklWEtX5MWPYJ-M/pub?start=true&loop=false&delayms=30000&slide=id.g19819563c7e\\_0\\_367](https://docs.google.com/presentation/d/e/2PACX-1vRiU7zISF3MzvpwF3S0lvGxis6mvi70IfBgU8dWByxy_RSJ4gHNo5WpgVOZYbcoXXklWEtX5MWPYJ-M/pub?start=true&loop=false&delayms=30000&slide=id.g19819563c7e_0_367)

Nobody can manipulate these functions such as changing the jackpot address or set the dividend address etc. apart from owners, we cannot allow that for obvious reasons."



## CONTRACT UNIQUENESS

Novoos contract also includes various fee properties, such as liquidity fee, improved reward fee, allowance fund fee, etc. There is a price recovery system, a dip reward system, anti-bot system, and an LP management system.

Novoos contract implements accumulative taxing system, ground zero protocol, advanced airdrop algorithm.

There are some properties marked as "for future use" such as the basic variables for allowance fund, bank, and Mariana trench, and fee properties for capital fee, allowance fund fee, bank fee, etc.

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL



Identifier	Definition	Severity
CEN-02	Initial asset distribution	Medium ●

All of the initially minted assets are sent to the project owner when deploying the contract. This can be an issue as the project owner can distribute tokens without consulting the community.

```
_mint(owner(), 1000000000 * (10 ** 18));
```

## RECOMMENDATION

Project must communicate with stakeholders and obtain the community consensus while distributing assets.


## RESOLUTION

Novoos distributes initially minted assets according to a pre-determined asset allocation model.

Review:

- 1) <https://docs.novoos.net/novoos-distribution/distribution-reasoning>
- 2) <https://docs.novoos.net/novoos-distribution/distribution-chart>
- 3) <https://docs.novoos.net/novoos-distribution/break-down>
- 4) <https://docs.novoos.net/novoos-distribution/vesting-and-locks>
- 5) <https://docs.novoos.net/novoos-distribution/allocation-and-distribution>



Identifier	Definition	Severity
CEN-04	Privileged role receiving LP tokens	Minor 

Smart contract function addLiquidity() sends liquidity to bankAddress.


```
function addLiquidity(uint256 tokenAmount, uint256 bnbAmount) private {
    // Approve token transfer to cover all possible scenarios
    _approve(address(this), address(uniswapV2Router), tokenAmount);

    // Add the liquidity
    uniswapV2Router.addLiquidityETH{value: bnbAmount}(
        address(this),
        tokenAmount,
        0, // Slippage is unavoidable
        0, // Slippage is unavoidable
        bankAddress,
        block.timestamp
    );
}
```

## RECOMMENDATION

Send LP tokens to dead address or unreachable address.



Identifier	Definition	Severity
NOV-01	Redundant privilege in <code>_setAutomatedMarketMakerPair</code>	Minor 

Privileged role can call `setAutomatedMarketMakerPair()` and `_setAutomatedMarketMakerPair()`.

```
function _setAutomatedMarketMakerPair(
    address pair,
    bool value
) private onlyOwner {
    require(
        automatedMarketMakerPairs[pair] != value,
        "Novoos: Automated market maker pair is already set to that value"
    );
    automatedMarketMakerPairs[pair] = value;
    if (value) {
        isDividendExempt[pair] = true;
    }
    emit SetAutomatedMarketMakerPair(pair, value);
}
```

## RECOMMENDATION

`_setAutomatedMarketMakerPair()` is an internal function. `onlyOwner` access control is redundant.



Identifier	Definition	Severity
CEN-09	Use of proxy and upgradeable contracts	Critical ●

Privileged role can initiate contract implementation. Contract upgradeability allows privileged roles to change current contract implementation.

## RECOMMENDATION


Test and validate current contract thoroughly before deployment. Future contract upgradeability negatively elevates centralization risk.

## RESOLUTION

According to Novoos, it has implemented upgradeable contracts for continuous seamless contract enhancements without any relaunches, and migrations.





Identifier	Definition	Severity
LOG-01	Lack of appropriate arbitrary boundaries	Minor 

Below mentioned functions are set without any arbitrary boundaries.

```
setWalletBalance()  
setMaxBuyTransaction()  
setMaxSellTransaction()  
updateGasForProcessing()  
setJackpotLimit()  
setTripSwitchDuration()  
setTripSwitchPeriod()
```

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL

## RECOMMENDATION

These functions should be provided appropriate upper and lower boundaries.



Identifier	Definition	Severity
COD-05	Missing zero address validation	

Below mentioned functions are missing zero address input validation:

```
updateUniswapV2Router()  
setAutomatedMarketMakerPair()  
setDividendDistributor()
```

## RECOMMENDATION

Validate if the modified address is dead(0) or not.

## ACKNOWLEDGEMENT

Novoos team has acknowledged to this finding, and agreed to keep it as-is.



Identifier	Definition	Severity
COD-06	Unknown externally owned account	Minor 

An externally owned account (EOA) has no code, and one can send messages from an externally owned account by creating and signing a transaction.

```
allowanceFundAddress = 0xbfe860f7f8d3b3f51B91Ae4757859c2Bb89C197E;
bankAddress = 0xf2e14c21ed6B505b4E51Bb494481378eE5C9Ca85;
marianatrenchAddress = 0x283cc27E7844A792ef4cBE7781b75a65596FC29A;
insuranceAndRewardFundAddress = 0x29B4920AA795899aefB0514BC273bb040F4AcDD4;
```

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL

## RECOMMENDATION

Private keys of externally owned accounts must be secured carefully.

## RESOLUTION

According to Novoos team, EOAs are backed up and secured adequately. Additionally, The Mariana Trench is a contract deployed with no ownership.



Identifier	Definition	Severity
COD-09	Lack of contract balance withdraw	

Smart contract may collect tokens, and ethers from external addresses. Some swap, and liquidity-add events may accumulate residual ethers, and tokens.

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL

## RECOMMENDATION

Add `withdraw()` function to take out tokens and ethers from the contract.



Identifier	Definition	Severity
COD-10	Third Party Dependencies	Unknown 🟠

Smart contract is interacting with third party protocols, contracts, market makers like Pancakeswap, Mariana Trench contract, OpenZeppelin tools, blockchain dependencies via block timestamps and numbers. The scope of the audit treats these entities as black boxes and assumes their functional correctness. However, in the real world, third parties can be compromised, and exploited. Moreover, upgrades in third parties can create severe impacts, e.g., increased transactional fees, deprecation of previous routers, etc.

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL

## RECOMMENDATION

Inspect third party dependencies regularly, and mitigate severe impacts whenever necessary.

## RESOLUTION

Novoos team will inspect third party dependencies regularly to mitigate down-times and vulnerabilities caused by external protocols, dependencies, contracts, decentralized applications, etc.



Identifier	Definition	Severity
VOL-01	Irrelevant code	Minor ●

Redundant code across the smart contract.


## RECOMMENDATION

Remove redundant and dead code.

## RESOLUTION

According to Novoos team, redundant code is to keep track of future reviews of the code and contract implementation upgrade.



Identifier	Definition	Severity
COM-01	Floating compiler status	Minor 

Compiler is set to

```
pragma solidity ^0.8.2;
```

INTERFI  
CONFIDENTIAL

INTERFI  
CONFIDENTIAL

## RECOMMENDATION

Pragma should be fixed to the version that you're indenting to deploy your contracts with.

## RESOLUTION

According to Novoos team, 0.8.2 was the stable version at the time of deployment. Future upgrades will have appropriate stable versions.



## DISCLAIMERS

InterFi Network provides the easy-to-understand audit of solidity source codes (commonly known as smart contracts).

The smart contract for this particular audit was analyzed for common contract vulnerabilities, and centralization exploits. This audit report makes no statements or warranties on the security of the code. This audit report does not provide any warranty or guarantee regarding the absolute bug-free nature of the smart contract analyzed, nor do they provide any indication of the client's business, business model or legal compliance. This audit report does not extend to the compiler layer, any other areas beyond the programming language, or other programming aspects that could present security risks. Cryptographic tokens are emergent technologies, they carry high levels of technical risks and uncertainty. 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. This audit report could include false positives, false negatives, and other unpredictable results.

## CONFIDENTIALITY

This report is subject to the terms and conditions (including without limitations, description of services, confidentiality, disclaimer and limitation of liability) outlined in the scope of the audit provided to the client. This report should not be transmitted, disclosed, referred to, or relied upon by any individual for any purpose without InterFi Network's prior written consent.

## NO FINANCIAL ADVICE

This audit report does not indicate the endorsement of any particular project or team, nor guarantees its security. No third party should rely on the reports in any way, including to make any decisions to buy or sell a product, service or any other asset. The information provided in this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. This audit report should not be used in any way





to make decisions around investment or involvement. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort.

FOR AVOIDANCE OF DOUBT, SERVICES, INCLUDING ANY ASSOCIATED AUDIT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

## **TECHNICAL DISCLAIMER**

ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, INTERFI NETWORK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO SERVICES, AUDIT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, INTERFI NETWORK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, INTERFI NETWORK MAKES NO WARRANTY OF ANY KIND THAT ALL SERVICES, AUDIT REPORTS, SMART CONTRACT AUDITS, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE CLIENT'S OR ANY OTHER INDIVIDUAL'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE.

## **TIMELINESS OF CONTENT**

The content contained in this audit report is subject to change without any prior notice. InterFi Network does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following the publication.



## LINKS TO OTHER WEBSITES

This audit report provides, through hypertext or other computer links, access to websites and social accounts operated by individuals other than InterFi Network. Such hyperlinks are provided for your reference and convenience only and are the exclusive responsibility of such websites' and social accounts' owners. You agree that InterFi Network is not responsible for the content or operation of such websites and social accounts and that InterFi Network shall have no liability to you or any other person or entity for the use of third-party websites and social accounts. You are solely responsible for determining the extent to which you may use any content at any other websites and social accounts to which you link from the report.

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI  
CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL



## ABOUT INTERFI NETWORK

InterFi Network provides intelligent blockchain solutions. We provide solidity development, testing, and auditing services. We have developed 150+ solidity codes, audited 1000+ smart contracts, and analyzed 500,000+ code lines. We have worked on major public blockchains e.g., Ethereum, Binance, Cronos, Doge, Polygon, Avalanche, Metis, Fantom, Bitcoin Cash, Velas, Oasis, etc.

InterFi Network is built by engineers, developers, UI experts, and blockchain enthusiasts. Our team currently consists of 4 core members, and 6+ casual contributors.

Website: <https://interfi.network>

Email: [hello@interfi.network](mailto:hello@interfi.network)

GitHub: <https://github.com/interfinetwork>

Telegram (Engineering): <https://t.me/interfiaudits>

Telegram (Onboarding): <https://t.me/interfisupport>



 interfinetwork

 hello@interfi.network

 <https://interfi.network>

SMART CONTRACT AUDITS | SOLIDITY DEVELOPMENT AND TESTING  
RELENTLESSLY SECURING PUBLIC AND PRIVATE BLOCKCHAINS