InterFi NETWORK

SMART CONTRACT AUDIT

interfinetwork

hello@interfi.network

https://interfi.network

PREPARED FOR

MINERSWAP

INTRODUCTION

Auditing Firm	InterFi Network
Client Firm	MinerSwap
Methodology	Automated Analysis, Manual Code Review
Language	Solidity
Contracts	Multiple Contracts
Blockchain	Binance Smart Chain
Centralization	Ownership is locked in a time-lock contract
Commit	bbff10b76a600ba69df970584cabae78d0b84e83
Website	https://minerswap.fi/
Telegram	https://t.me/minerswapfi
Twitter	https://twitter.com/minerswapfi
Discord	https://discord.gg/kPmsWNDarq
Prelim Analysis Date	October 22, 2022
Final Report Date	October 28, 2022

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

ℹ Verify the authenticity of this report on our website: https://www.interfi.network/audits

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 🔴	Major 🟠	Medium 🟡	Minor 🟢	Unknown 🟤
Open	0	0	0	3	0
Acknowledged	1	0	0	4	0
Resolved	0	1	1	0	0
Noteworthy Privileges	Refer to PAGE 28 for centralization related privileges				

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

ℹ️ 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.

ℹ️ 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 7

RISK CATEGORIES 9

CENTRALIZED PRIVILEGES 10

AUTOMATED ANALYSIS 11

INHERITANCE GRAPH 27

MANUAL REVIEW 28

DISCLAIMERS 41

ABOUT INTERFI NETWORK 44

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

SCOPE OF WORK

InterFi was consulted by MinerSwap 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:

- IMinerSwapCore.sol
- IMinerSwapFactory.sol
- IMinerSwapPair.sol
- MinerSwapCore.sol
- MinerSwapFactory.sol
- MinerSwapRouter.sol
- MinerPool.sol
- Token.sol
- Timelock.sol

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTER
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

ℹ️ 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:

MinerSwapCore.sol #0xE085BDd4D0b537a91Dba4A6d3371123E4EEB3Ab8
https://uniwscan.com/address/0xE085BDd4D0b537a91Dba4A6d3371123E4EEB3Ab8/contracts#address-tabs

MinerSwapFactory #0x91836D77AF0A5fdA36C5a1F3c11DbC7766dE4D03
https://uniwscan.com/address/0x91836D77AF0A5fdA36C5a1F3c11DbC7766dE4D03/contracts#address-tabs

MinerSwapRouter #0x8f048e25fEcd0AFdE2af78d9966C010D94e76aEc
https://uniwscan.com/address/0x8f048e25fEcd0AFdE2af78d9966C010D94e76aEc/contracts#address-tabs

MinerPool #0x710a5dEE58a1845E1f17FC8fBF583FE0Ab29c021
https://uniwscan.com/address/0x710a5dEE58a1845E1f17FC8fBF583FE0Ab29c021/contracts#address-tabs

MinerSwap Token #0x5Ee9Cb7AEC566bccF3b24C6C96744972878292d3
https://uniwscan.com/address/0x5Ee9Cb7AEC566bccF3b24C6C96744972878292d3/contracts#address-tabs

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

Timelock #0x91aEd49c1479FcF9D2a960bf65ABE2635C51e058
https://uniwscan.com/address/0x91aEd49c1479FcF9D2a960bf65ABE2635C51e058/contracts#address-tabs

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	○ Token Supply Manipulation ○ Access Control and Authorization ○ Assets Manipulation ○ Ownership Control ○ Liquidity Access ○ Stop and Pause Trading ○ Ownable Library Verification

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

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

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

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.

ℹ️ 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.

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

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.

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

ℹ️ 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

IMinerSwapCore	Interface			
└	initialize	External !	🛑	NO!
└	factory	External !		NO!
└	feeToSetter	External !		NO!
└	vaultController	External !		NO!
└	WETH	External !		NO!
└	Stable	External !		NO!
└	ethPrice	External !		NO!
└	ethStablePair	External !		NO!
└	feePoint	External !		NO!
└	initPair	External !	🛑	NO!
└	mint	External !	🛑	NO!
└	emitMint	External !	🛑	NO!
└	burn	External !	🛑	NO!
└	emitBurn	External !	🛑	NO!
└	swap	External !	🛑	NO!
└	emitSwap	External !	🛑	NO!
└	emitTransfer	External !	🛑	NO!
└	skim	External !	🛑	NO!
└	sync	External !	🛑	NO!

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

└	getFees	External ❗		NO❗
└	getPairType	External ❗		NO❗
└	getCumulativePrice	External ❗		NO❗
└	getKlast	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
└	getPair	External ❗		NO❗
└	getPairTokens	External ❗		NO❗
└	getPairLength	External ❗		NO❗
└	getEncodedRates	External ❗		NO❗
└	getEffectiveRates	External ❗		NO❗
└	getEffectivePrices	External ❗		NO❗
└	getPairPrices	External ❗		NO❗
└	getTracking	External ❗		NO❗
└	getBlacklisted	External ❗		NO❗
||||||				
IMinerSwapFactory	Interface	|||		
└	INIT_CODE_PAIR_HASH	External ❗		NO❗
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
||||||				
IMinerSwapPair	Interface	|||		
└	name	External ❗		NO❗

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

└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	getPairInfo	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	mint	External ❗	🛑	NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗
└	checkVault	External ❗	🛑	NO❗

|||||

IERC20	Interface				
└	name	External ❗		NO❗

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IMinerSwapCallee	Interface			
└	minerSwapCall	External ❗	🛑	NO❗
IController	Interface			
└	depositPair	External ❗	🛑	NO❗
└	withdrawAll	External ❗	🛑	NO❗
Math	Library			
└	min	Internal 🔒		
└	sqrt	Internal 🔒		
EnumerableSet	Library			
└	_add	Private 🔐	🛑	
└	_remove	Private 🔐	🛑	
└	_contains	Private 🔐		
└	_length	Private 🔐		
└	_at	Private 🔐		
└	_values	Private 🔐		
└	add	Internal 🔒	🛑	
└	remove	Internal 🔒	🛑	
└	contains	Internal 🔒		
└	length	Internal 🔒		
└	at	Internal 🔒		

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 CONFIDEN

└	values	Internal 🔒		
└	add	Internal 🔒	🛑	
└	remove	Internal 🔒	🛑	
└	contains	Internal 🔒		
└	length	Internal 🔒		
└	at	Internal 🔒		
└	values	Internal 🔒		
└	add	Internal 🔒	🛑	
└	remove	Internal 🔒	🛑	
└	contains	Internal 🔒		
└	length	Internal 🔒		
└	at	Internal 🔒		
└	values	Internal 🔒		
SafeTransferLib	Library			
└	safeTransferETH	Internal 🔒	🛑	
└	safeTransferFrom	Internal 🔒	🛑	
└	safeTransfer	Internal 🔒	🛑	
└	safeApprove	Internal 🔒	🛑	
UQ112x112	Library			
└	encode	Internal 🔒		
└	uqdiv	Internal 🔒		
MinerSwapERC20	Implementation	IERC20		
└	updateDomainSeparator	Internal 🔒	🛑	
└	name	Public ❗	NO❗	
└	symbol	Public ❗	NO❗	
└	decimals	Public ❗	NO❗	
└	totalSupply	Public ❗	NO❗	
└	balanceOf	Public ❗	NO❗	
└	allowance	Public ❗	NO❗	

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

└	nonces	Public ❗		NO❗
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	approve	Public ❗	🛑	NO❗
└	transfer	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	permit	Public ❗	🛑	NO❗
MinerSwapPair	Implementation	IMinerSwapPair, MinerSwapERC20		
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	getPairInfo	External ❗		NO❗
└	initialize	External ❗	🛑	NO❗
└	_update	Private 🔐	🛑	
└	mint	External ❗	🛑	lock
└	burn	External ❗	🛑	lock
└	swap	External ❗	🛑	lock
└	skim	External ❗	🛑	lock
└	sync	External ❗	🛑	lock
└	checkVault	External ❗	🛑	onlyCore
└	name	Public ❗		NO❗
└	symbol	Public ❗		NO❗
└	decimals	Public ❗		NO❗
└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗

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

└	transfer	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	nonces	Public ❗		NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	permit	Public ❗	🛑	NO❗
MinerSwapFactory	Implementation	IMinerSwapFactory		
└	initialize	External ❗	🛑	NO❗
└	allPairsLength	External ❗		NO❗
└	feeTo	External ❗		NO❗
└	INIT_CODE_PAIR_HASH	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
MinerSwapCore	Implementation	IMinerSwapCore		
└	newToken	Internal 🔒	🛑	
└	initialize	External ❗	🛑	NO❗
└	initPair	External ❗	🛑	onlyFactory
└	updateEffectiveRates	Internal 🔒	🛑	
└	_update	Private 🔐	🛑	
└	mint	External ❗	🛑	lock
└	emitMint	External ❗	🛑	NO❗
└	burn	External ❗	🛑	lock
└	emitBurn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	lock
└	emitSwap	External ❗	🛑	NO❗
└	emitTransfer	External ❗	🛑	NO❗
└	skim	External ❗	🛑	lock
└	sync	External ❗	🛑	lock

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

└	getBalance	Internal 🔒		
└	getValueDept	External ❗		NO❗
└	depositVault	Internal 🔒	🛑	
└	withdrawVault	Internal 🔒	🛑	
└	depositExternal	External ❗	🛑	onlySetter
└	withdrawExternal	External ❗	🛑	onlySetter
└	checkAmount	Internal 🔒	🛑	
└	reDeposit	Internal 🔒	🛑	
└	sortTokens	Internal 🔒		
└	pairFor	Internal 🔒		
└	getPair	External ❗		NO❗
└	getPairType	External ❗		NO❗
└	getPairTokens	External ❗		NO❗
└	getPairLength	External ❗		NO❗
└	getEncodedRates	Public ❗		NO❗
└	getEffectiveRates	Public ❗		NO❗
└	getEffectivePrices	Public ❗		NO❗
└	getPairPrices	External ❗		NO❗
└	getCumulativePrice	External ❗		NO❗
└	getTracking	External ❗		NO❗
└	getBlacklisted	External ❗		NO❗
└	getKlast	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	_f	Internal 🔒		
└	_d	Internal 🔒		
└	_get_y	Internal 🔒		
└	sCurve	Internal 🔒		
└	verify	Internal 🔒		
└	getFees	Public ❗		NO❗
└	collectFee	Internal 🔒	🛑	
└	_getAmountOut	Internal 🔒		

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

└	_getAmountIn	Internal 🔒		
└	getAmountOut	External ❗		NO❗
└	getAmountIn	External ❗		NO❗
└	getAmountsOut	External ❗		NO❗
└	getAmountsIn	External ❗		NO❗
└	updatePairType	External ❗	🛑	onlySetter
└	setFeePoint	External ❗	🛑	onlySetter
└	setTokenFeePoint	External ❗	🛑	onlySetter
└	setPairFeeAccount	External ❗	🛑	onlySetter
└	setVaultController	External ❗	🛑	onlySetter
└	setFeeToSetter	External ❗	🛑	onlySetter
└	updateStableToken	External ❗	🛑	onlySetter
└	addTracking	External ❗	🛑	onlySetter
└	removeTracking	External ❗	🛑	onlySetter
└	addBlacklist	External ❗	🛑	onlySetter
└	removeBlacklist	External ❗	🛑	onlySetter
└	setMinimumLiq	External ❗	🛑	onlySetter
└	transfer	External ❗	🛑	onlySetter

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

SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	mod	Internal 🔒		

Timelock	Implementation				
└	initialize	External ❗	🛑	NO❗

| └ | <Receive Ether> | External ❗ | 💵 |NO❗ |

| └ | setDelay | Public ❗ | 🛑 |NO❗ |

| └ | acceptAdmin | Public ❗ | 🛑 |NO❗ |

| └ | setPendingAdmin | Public ❗ | 🛑 |NO❗ |

| └ | queueTransaction | Public ❗ | 🛑 |NO❗ |

| └ | cancelTransaction | Public ❗ | 🛑 |NO❗ |

| └ | executeTransaction | Public ❗ | 💵 |NO❗ |

| └ | getBlockTimestamp | Internal 🔒 | | |

| **SafeMath** | Library | |||

| └ | tryAdd | Internal 🔒 | | |

| └ | trySub | Internal 🔒 | | |

| └ | tryMul | Internal 🔒 | | |

| └ | tryDiv | Internal 🔒 | | |

| └ | tryMod | Internal 🔒 | | |

| └ | add | Internal 🔒 | | |

| └ | sub | Internal 🔒 | | |

| └ | mul | Internal 🔒 | | |

| └ | div | Internal 🔒 | | |

| └ | mod | Internal 🔒 | | |

| └ | sub | Internal 🔒 | | |

| └ | div | Internal 🔒 | | |

| └ | mod | Internal 🔒 | | |

||||||

| **EnumerableSet** | Library | |||

| └ | _add | Private 🔐 | 🛑 | |

| └ | _remove | Private 🔐 | 🛑 | |

| └ | _contains | Private 🔐 | | |

| └ | _length | Private 🔐 | | |

| └ | _at | Private 🔐 | | |

| └ | _values | Private 🔐 | | |

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

└	add	Internal 🔒	🛑	
└	remove	Internal 🔒	🛑	
└	contains	Internal 🔒		
└	length	Internal 🔒		
└	at	Internal 🔒		
└	values	Internal 🔒		
└	add	Internal 🔒	🛑	
└	remove	Internal 🔒	🛑	
└	contains	Internal 🔒		
└	length	Internal 🔒		
└	at	Internal 🔒		
└	values	Internal 🔒		
└	add	Internal 🔒	🛑	
└	remove	Internal 🔒	🛑	
└	contains	Internal 🔒		
└	length	Internal 🔒		
└	at	Internal 🔒		
└	values	Internal 🔒		
||||||				
Context	Implementation		|||	
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
||||||				
Ownable	Implementation	Context |||		
└	<Constructor>	Public ❗	🛑	|NO❗ |
└	owner	Public ❗		|NO❗ |
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
||||||				
Address	Library		|||	

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	verifyCallResult	Internal 🔒		
IERC20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
IERC20Permit	Interface			
└	permit	External ❗	🛑	NO❗
└	nonces	External ❗		NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
SafeERC20	Library			
└	safeTransfer	Internal 🔒	🛑	
└	safeTransferFrom	Internal 🔒	🛑	
└	safeApprove	Internal 🔒	🛑	
└	safeIncreaseAllowance	Internal 🔒	🛑	
└	safeDecreaseAllowance	Internal 🔒	🛑	
└	safePermit	Internal 🔒	🛑	

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

| └ | _callOptionalReturn | Private 🔐 | 🛑 | | |

||||||

| **IMinerSwap** | Interface | | | |

| └ | mint | External ❗ | 🛑 | NO❗ |

||||||

| **MasterChef** | Implementation | Ownable | | |

| └ | initialize | External ❗ | 🛑 | NO❗ |

| └ | poolLength | External ❗ | | NO❗ |

| └ | add | External ❗ | 🛑 | onlyOwner |

| └ | set | External ❗ | 🛑 | onlyOwner |

| └ | getMultiplier | Public ❗ | | NO❗ |

| └ | pendingMsp | External ❗ | | NO❗ |

| └ | massUpdatePools | Public ❗ | 🛑 | NO❗ |

| └ | updatePool | Internal 🔒 | 🛑 | |

| └ | deposit | External ❗ | 🛑 | NO❗ |

| └ | withdraw | External ❗ | 🛑 | NO❗ |

| └ | emergencyWithdraw | External ❗ | 🛑 | NO❗ |

| └ | setDevAddress | External ❗ | 🛑 | NO❗ |

| └ | setFeeAddress | External ❗ | 🛑 | NO❗ |

| └ | setStartBlock | External ❗ | 🛑 | NO❗ |

| └ | setMultiplier | External ❗ | 🛑 | onlyOwner |

| └ | setDevRate | External ❗ | 🛑 | onlyOwner |

| └ | updateEmissionRate | External ❗ | 🛑 | onlyOwner |

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

| **AddressUpgradeable** | Library | | | |

| └ | isContract | Internal 🔒 | | |

| └ | sendValue | Internal 🔒 | 🛑 | |

| └ | functionCall | Internal 🔒 | 🛑 | |

| └ | functionCall | Internal 🔒 | 🛑 | |

| └ | functionCallWithValue | Internal 🔒 | 🛑 | |

| └ | functionCallWithValue | Internal 🔒 | 🛑 | |

| └ | functionStaticCall | Internal 🔒 | | |

| └ | functionStaticCall | Internal 🔒 | | |

| └ | verifyCallResult | Internal 🔒 | | |

||||||

| **Initializable** | Implementation | |||

| └ | _disableInitializers | Internal 🔒 | 🛑 | |

||||||

| **ContextUpgradeable** | Implementation | Initializable |||

| └ | __Context_init | Internal 🔒 | 🛑 | onlyInitializing |

| └ | __Context_init_unchained | Internal 🔒 | 🛑 | onlyInitializing |

| └ | _msgSender | Internal 🔒 | | |

| └ | _msgData | Internal 🔒 | | |

||||||

| **IERC20Upgradeable** | Interface | |||

| └ | totalSupply | External ❗ | |NO❗ |

| └ | balanceOf | External ❗ | |NO❗ |

| └ | transfer | External ❗ | 🛑 |NO❗ |

| └ | allowance | External ❗ | |NO❗ |

| └ | approve | External ❗ | 🛑 |NO❗ |

| └ | transferFrom | External ❗ | 🛑 |NO❗ |

||||||

| **IERC20MetadataUpgradeable** | Interface | IERC20Upgradeable |||

| └ | name | External ❗ | |NO❗ |

| └ | symbol | External ❗ | |NO❗ |

| └ | decimals | External ❗ | |NO❗ |

||||||

| **ERC20Upgradeable** | Implementation | Initializable, ContextUpgradeable, IERC20Upgradeable, IERC20MetadataUpgradeable |||

| └ | __ERC20_init | Internal 🔒 | 🛑 | onlyInitializing |

| └ | __ERC20_init_unchained | Internal 🔒 | 🛑 | onlyInitializing |

| └ | name | Public ❗ | |NO❗ |

| └ | symbol | Public ❗ | |NO❗ |

| └ | decimals | Public ❗ | |NO❗ |

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

└	totalSupply	Public ❗		NO❗
└	balanceOf	Public ❗		NO❗
└	transfer	Public ❗	🛑	NO❗
└	allowance	Public ❗		NO❗
└	approve	Public ❗	🛑	NO❗
└	transferFrom	Public ❗	🛑	NO❗
└	increaseAllowance	Public ❗	🛑	NO❗
└	decreaseAllowance	Public ❗	🛑	NO❗
└	_transfer	Internal 🔒	🛑	
└	_mint	Internal 🔒	🛑	
└	_burn	Internal 🔒	🛑	
└	_approve	Internal 🔒	🛑	
└	_spendAllowance	Internal 🔒	🛑	
└	_beforeTokenTransfer	Internal 🔒	🛑	
└	_afterTokenTransfer	Internal 🔒	🛑	
OwnableUpgradeable	Implementation	Initializable, ContextUpgradeable		
└	__Ownable_init	Internal 🔒	🛑	onlyInitializing
└	__Ownable_init_unchained	Internal 🔒	🛑	onlyInitializing
└	owner	Public ❗		NO❗
└	_checkOwner	Internal 🔒		
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_transferOwnership	Internal 🔒	🛑	
EnumerableSetUpgradeable	Library			
└	_add	Private 🔐	🛑	
└	_remove	Private 🔐	🛑	
└	_contains	Private 🔐		
└	_length	Private 🔐		
└	_at	Private 🔐		

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

| └ | _values | Private 🔐 | | |

| └ | add | Internal 🔒 | 🛑 | |

| └ | remove | Internal 🔒 | 🛑 | |

| └ | contains | Internal 🔒 | | |

| └ | length | Internal 🔒 | | |

| └ | at | Internal 🔒 | | |

| └ | values | Internal 🔒 | | |

| └ | add | Internal 🔒 | 🛑 | |

| └ | remove | Internal 🔒 | 🛑 | |

| └ | contains | Internal 🔒 | | |

| └ | length | Internal 🔒 | | |

| └ | at | Internal 🔒 | | |

| └ | values | Internal 🔒 | | |

| └ | add | Internal 🔒 | 🛑 | |

| └ | remove | Internal 🔒 | 🛑 | |

INTERFI CONFIDENTIAL AUDIT REPORT

| └ | contains | Internal 🔒 | | |

| └ | length | Internal 🔒 | | |

| └ | at | Internal 🔒 | | |

| └ | values | Internal 🔒 | | |

||||||

| **Token** | Implementation | ERC20Upgradeable, OwnableUpgradeable |||

| └ | initialize | External ❗️ | 🛑 | initializer |

| └ | addMinter | External ❗️ | 🛑 | onlyOwner |

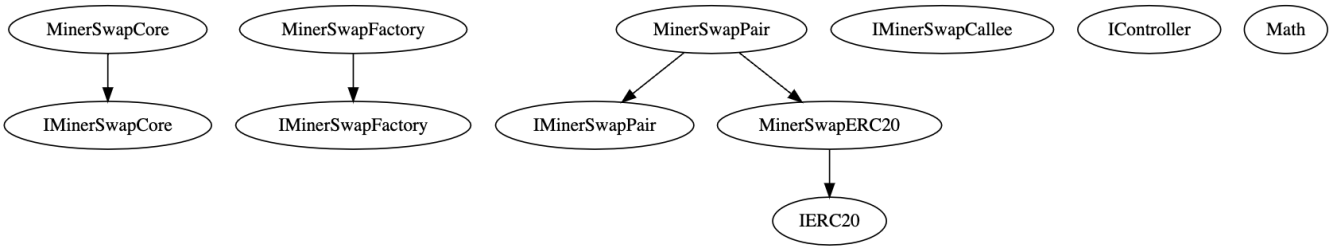
| └ | removeMinter | External ❗️ | 🛑 | onlyOwner |

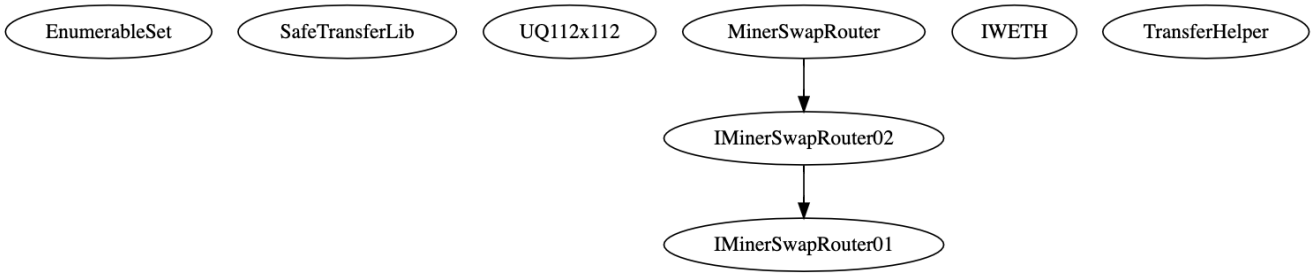
| └ | getMinters | External ❗️ | |NO❗️ |

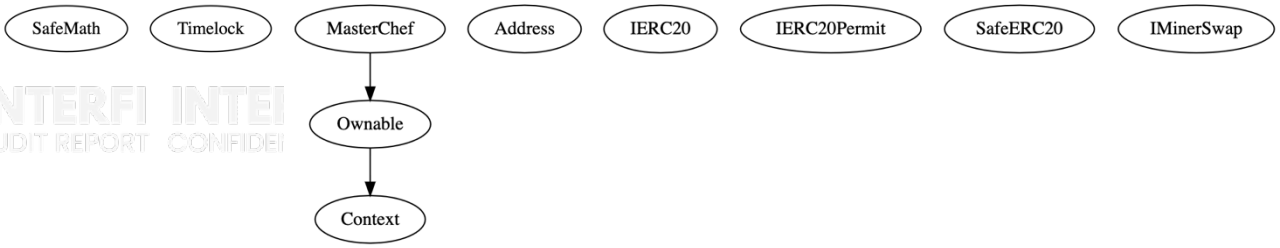
| └ | mintersCount | External ❗️ | |NO❗️ |

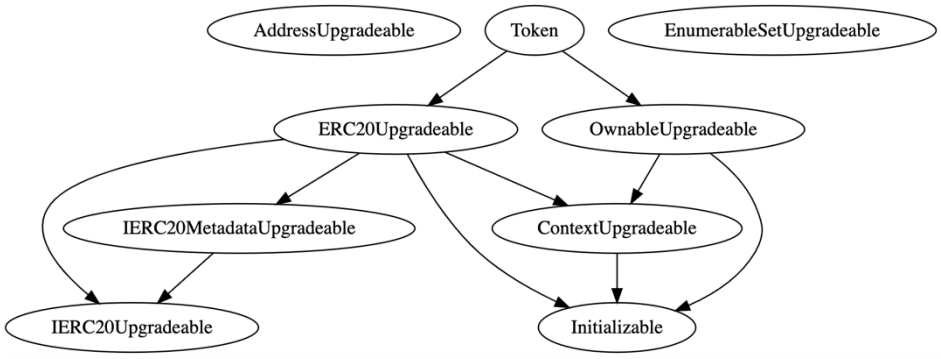
| └ | mint | External ❗️ | 🛑 | onlyMinter |

INHERITANCE GRAPH









INTERFI INTERFI INTEI
NFIDENTIAL AUDIT REPORT CONFIDEI
FI INTER
RT CONFIDEN

MANUAL REVIEW

Identifier	Definition	Severity
CEN-01	Centralization privileges of MinerSwap	Major 🟠
CEN-07	Authorizations and access controls	Minor 🟢

- Smart contract MinerSwapPair.sol sets `onlyFactory` and `onlyCore` modifiers.

```
modifier onlyFactory() {
    require(msg.sender == factory, 'Not factory');
modifier onlyCore() {
    require(msg.sender == address(core), 'Not core');

   initPair onlyFactory
   checkVault onlyCore
```

INTERFI CONFIDENTIAL AUDIT REPORT

- Smart contract MinerSwapCore.sol sets `onlyFactory` and `onlySetter` modifiers.

```
modifier onlyFactory() {
    require(msg.sender == factory, 'Not factory');
modifier onlySetter() {
    require(msg.sender == feeToSetter, 'Not feeToSetter');

   depositExternal onlySetter
   withdrawExternal onlySetter
```

- Smart contract MinerPool.sol sets `onlyOwner` modifier.

```
modifier onlyOwner() {
    _checkOwner();

   renounceOwnership onlyOwner
   transferOwnership onlyOwner
   add onlyOwner
   set onlyOwner
   updateEmissionRate onlyOwner
```

- Smart contract Token.sol sets `onlyOwner` and `onlyMinter` modifiers.

```
modifier onlyOwner() {
    _checkOwner();
modifier onlyMinter() {
    require(minters.contains(_msgSender()), "FORBIDDEN");
```

`renounceOwnership` `onlyOwner`
`transferOwnership` `onlyOwner`
`initialize` `onlyOwner`
`addMinter` `onlyOwner`
`removeMinter` `onlyOwner`
`mint` `onlyMinter`

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTER
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

RECOMMENDATION

Private keys of all privileged roles must be secured carefully. Please refer to PAGE-09 CENTRALIZED PRIVILEGES for a detailed understanding.

RESOLUTION

Smart contract ownership is forwarded to a time-lock contract.

Ownership transfers:

https://uniwscan.com/tx/0xfe2646416957805aaa73248bb1ee16d5b173d41a96c155f5d88118521ee954b3

https://uniwscan.com/tx/0x116ddaf8d22bfcb1066381c56b701b2b3b8ca337266848f0ee7948609ddf9292

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.

```
contract Token is ERC20Upgradeable, OwnableUpgradeable {
    using EnumerableSetUpgradeable for EnumerableSetUpgradeable.AddressSet;


    function initialize(string memory name_, string memory symbol_, uint8 decimal_, address
owner_) external initializer {
        __ERC20_init(name_, symbol_, decimal_);
        _transferOwnership(owner_);
        minters.add(owner_);
        emit NewMiner(owner_);
        emit Initialize(name_, symbol_, decimal_);
    }
```

TERFI NFIDENTIAL INTER CONFIDEN

RECOMMENDATION

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

ALLEVIATION

Smart contract ownership is forwarded to a time-lock contract.

Ownership transfers:

https://uniwscan.com/tx/0xfe2646416957805aaa73248bb1ee16d5b173d41a96c155f5d88118521ee954b3

https://uniwscan.com/tx/0x116ddaf8d22bfcb1066381c56b701b2b3b8ca337266848f0ee7948609ddf9292

Identifier	Definition	Severity
CEN-11	Privileged role performing mint	Medium 🟡

In smart contract Token.sol, Privileged role can mint assets.

```
function mint(address _to, uint256 _amount) external onlyMinter {
    _mint(_to, _amount);
    emit Mint(_to, _amount);
}
```

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

RECOMMENDATION

Declare and lock total asset supply. Access to mint function negatively elevates centralization risk.

RESOLUTION

Smart contract ownership is forwarded to a time-lock contract.

Ownership transfers:

https://uniwscan.com/tx/0xfe2646416957805aaa73248bb1ee16d5b173d41a96c155f5d88118521ee954b3

https://uniwscan.com/tx/0x116ddaf8d22bfcb1066381c56b701b2b3b8ca337266848f0ee7948609ddf9292

Identifier	Definition	Severity
LOG-02	Potential sandwich attack	Minor 🟢

Potential sandwich attack happens 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 front-running a transaction to purchase assets and make profits by back-running a transaction to sell assets. Below mentioned functions are called without setting restrictions on slippage or minimum output:

```
addLiquidity()
swapExactTokensForTokens()
swapTokensForExactTokens()
swapExactETHForTokens()
swapTokensForExactETH()
swapExactTokensForETH()
swapETHForExactTokens()
swapExactTokensForTokensSupportingFeeOnTransferTokens()
swapExactETHForTokensSupportingFeeOnTransferTokens()
swapExactTokensForETHSupportingFeeOnTransferTokens()
```

TERFI NFIDENTIAL INTER CONFIDEN

RECOMMENDATION

These functions should be provided reasonable minimum output amounts, instead of zero. Read more: https://coinmarketcap.com/alexandria/article/what-are-sandwich-attacks-in-defi-and-how-can-you-avoid-them

Identifier	Definition	Severity
LOG-03	Missing important checks	Minor 🟢

Below mentioned functions are missing parameter validation.

```
updatePool()
deposit()
emergencyWithdraw()
```

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTER
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

RECOMMENDATION

Add `require` checks to ensure that the token of the given pool is valid.

Identifier	Definition	Severity
LOG-04	Missing verification of checks-effects-interactions method	Minor 🟢

In function `add()`, `_lpToken` is pointing to a smart contract that is implemented based on ERC20. This smart contract can only be passed into function `add()` by owner as one of the parameters while the implementation of `_lpToken` is unknown statically, even if `_lpToken` followed the ERC20.

```
    function add(uint256 _allocPoint, IERC20 _lpToken, uint16 _depositFeeBP, bool
_withUpdate) public onlyOwner {
        require(!_poolAddresses.contains(address(_lpToken)), "add: duplicate pool");
        require(_depositFeeBP <= 10000, "add: invalid deposit fee basis points");
        if (_withUpdate) {
            massUpdatePools();
        }
        uint256 lastRewardBlock = block.number > startBlock ? block.number : startBlock;
        totalAllocPoint = totalAllocPoint.add(_allocPoint);
        poolInfo.push(PoolInfo({
            lpToken: _lpToken,
            allocPoint: _allocPoint,
            lastRewardBlock: lastRewardBlock,
            accMspPerShare: 0,
            depositFeeBP: _depositFeeBP
        }));
        _poolAddresses.add(address(_lpToken));
    }
```

TERFI NFIDENTIAL INTER CONFIDEN

RECOMMENDATION

Implement checks-effects-interactions verification method. Review https://fravoll.github.io/solidity-patterns/checks_effects_interactions.html for more information.

Identifier	Definition	Severity
COD-01	Use of `.call`	Minor 🟢

`.call()` used across smart contracts.

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTER
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

RECOMMENDATION

Avoid using `.call()` whenever possible when executing another contract function as it bypasses type checking, function existence check, and argument packing.

Identifier	Definition	Severity
COD-02	Timestamp manipulation via `block.timestamp` Avoid using `block.number` as timestamp	Minor 🟢

Be aware that the timestamp of the block can be manipulated by a miner. When the contract uses the timestamp to seed a random number, the miner can actually post a timestamp within 15 seconds of the block being validated, effectively allowing the miner to precompute an option more favorable to their chances, this is a critical exploit for contracts calculating random numbers, e.g., lottery.

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

RECOMMENDATION

To maintain block integrity, follow 15 seconds rule, and scale time dependent events accordingly.

Identifier	Definition	Severity
COD-04	Unclear error messages	

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTER
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

RECOMMENDATION

Provide accurate information strings for `require` related errors.

Identifier	Definition	Severity
COD-05	Missing zero address validation	

Validate if the input EOA is zero or not. Missing zero address validations across smart contracts.

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

RECOMMENDATION

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

Identifier	Definition	Severity
COD-11	Missing emit events	Minor 🟢

MasterChef.sol

Add missing emit events to `dev(), setFeeAddress(), updateEmissionRate().`

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

RECOMMENDATION

Add events for important functions, and emit them.

Identifier	Definition	Severity
COM-01	Floating compiler status	

Compiler is set to `^0.8.17`

TERFI
NFIDENTIAL
INTER
CONFIDEN

RECOMMENDATION

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

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.

INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTERFI INTER
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

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.

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

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 INTER
NFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDENTIAL AUDIT REPORT CONFIDEN

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

GitHub: https://github.com/interfinetwork

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

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