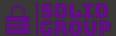


# AUDIT REPORT DATE APRIL 1ST FOR THEFINANCELABS.FINANCE PROJECT



Solid Group Auditing Service Telegram: @solid\_1

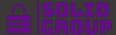
Twitter: https://twitter.com/solid\_group\_1



# Disclaimer

SolidGroup reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team.

SolidGroup Audits do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors. SolidGroup Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort. SolidGroup Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. SolidGroup's position is that each company and individual are responsible for their own due diligence and continuous security. SolidGroup in no way claims any guarantee of security or functionality of the technology we agree to analyze.



# Overview

Network: on-chain Binance Smart Chain

Website: http://thelab.finance

Twitter Group: <a href="https://twitter.com/thelabfinance">https://twitter.com/thelabfinance</a>
Telegram Group: <a href="https://t.me/thelabfinance">https://t.me/thelabfinance</a>

# Description

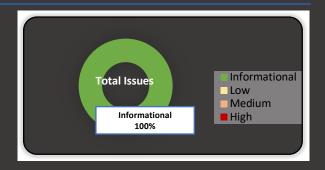
BSC Farming Protocol with Governance & Profit-Generating NFT's. A community-centered BSC farming solution with NTF's which will work as a Revenue Generator and as a Governance Model.

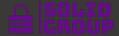
# Files In Scope

Contract Name	Address	MD5
LaboToken.sol	0x171401a3d18B21BFa3f9bF4F9637F36911583 65A	7AB826726F27C6C658D57C6DCB52D636
MasterChefV2.so	0x30f4cb706e65ABB3cbC3fFC2805E8Ff50eA8fb C8	9E491175F2716D76E25C71D2A4F4 FF59
Timelock.sol	0x0A974345327c8A7d7FC2e5eD6e2DE6343500 660E	05B3D7599449B2A4194082524009 95DF

# **Vulnerability Summary**

<ul><li>Informational severity Issues</li></ul>	3
<ul><li>Low severity issues</li></ul>	0
<ul><li>Medium severity issues</li></ul>	0
High severity issues	0

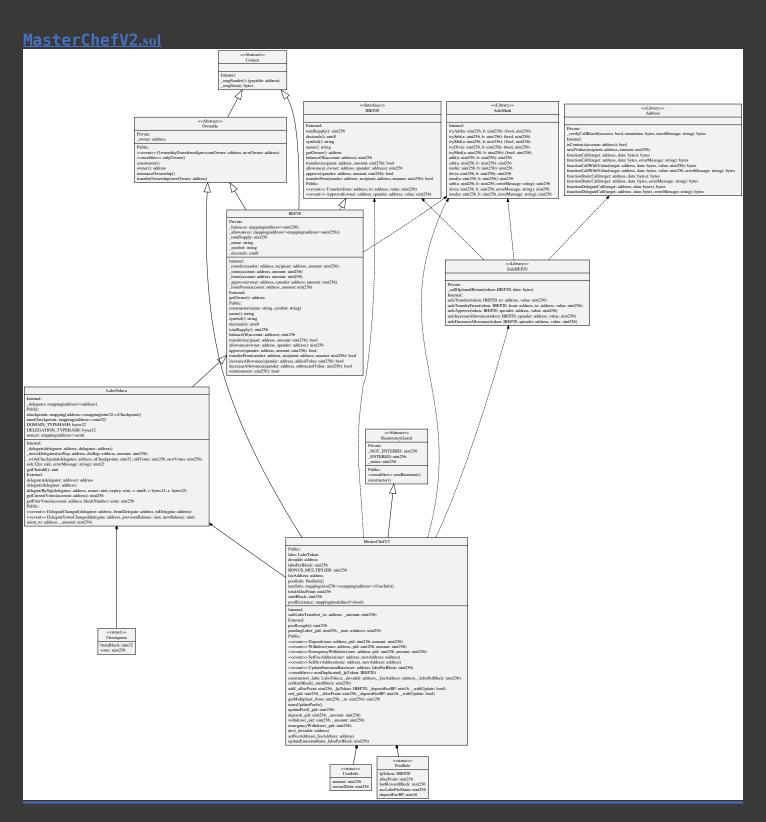


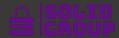


# **UML**

# LaboToken.sol <<Abstract>> Context Internal: \_msgSender(): (payable: address) \_msgData(): bytes <<Interface>> <<Library>> SafeMath IBEP20 <<Abstract>> Ownable Internal: tryAdd(a: uint256, b: uint256): (bool, uint256) trySub(a: uint256, b: uint256): (bool, uint256) trySub(a: uint256, b: uint256): (bool, uint256) tryDiv(a: uint256, b: uint256): (bool, uint256) tryDiv(a: uint256, b: uint256): (bool, uint256) tryDiv(a: uint256, b: uint256): (bool, uint256) add(a: uint256, b: uint256): uint256 sub(a: uint256, b: uint256): uint256 mul(a: uint256, b: uint256): uint256 div(a: uint256, b: uint256): uint256 mod(a: uint256, b: uint256; uint256 add(a: uint256, b: uint256; uint256 mod(a: uint256, b: uint256, uint256 mod(a: uint256, b: uint256, uint256; uint256 mod(a: uint256, b: uint256, uint256; uint25 External: totalSupply(): uint256 decimals(): uint8 symbol(): string Private: \_owner: address name(): string <event>> OwnershipTransferred(previousOwner: address, newOwner: address) getOwner(): address balanceOf(account; address); uint256 <<modifier>> onlyOwner() balance()f(account: address): uint256 transfer(recipient: address, amount: uint256): bool allowance(\_owner: address, spender: address): uint256 approve(spender: address, amount: uint256): bool transferfrom(sender: address, recipient: address, amount: uint256): bool Public: constructor() owner(): address renounceOwnership() transferOwnership(newOwner: address) Public: <event>> Transfer(from: address, to: address, value: uint256) <event>> Approval(owner: address, spender: address, value: uint256) BEP20 \_balances: mapping(address=>uint256) \_allowances: mapping(address=>mapping(address=>uint256)) \_totalSupply: uint256 name: string \_symbol: string \_decimals: uint8 Internal: transfer/sender: address, recipient: address, amount: uint256) \_mint(account: address, amount: uint256) \_burn(account: address, amount: uint256) \_approve(owner: address, spender: address, amount: uint256) \_burnFrom(account: address, amount: uint256) External: External: getOwner(): address Public: constructor(name: string, symbol: string) constructor(name: string, symbol: string) name(): string symbol(): string decimals(): uint8 totalSupply(): uint256 balanceOf(account: address): uint256 transfer(recipient: address, amount: uint256): bool allowance(owner: address, amount: uint256): bool approve(spender: address, amount: uint256): bool transferFrom(sender: address, amount: uint256): bool increaseAllowance(spender: address, addedValue: uint256): bool decreaseAllowance(spender: address, addedValue: uint256): bool mint(amount: uint256): bool LaboToken Internal: Internal: \_delegates: mapping(address=>address) Public: cheekpoints: mapping(address=>mapping(uint32=>Cheekpoint)) numCheekpoints: mapping(address=>uint32) DOMAIN\_TYPEHASH: byte332 DELEGATION\_TYPEHASH: byte332 DELEGATION\_TYPEHASH: byte312 nonces: mapping(address=>uint) Internal: delegates(delegator: address): address delegaticelegate: address), aduress/, aduress/ gerrior voies(account: aduress, nioexxumber: unit; unit250 Public: <event>> DelegateChanged(delegator: address, fromDelegate: address, toDelegate: address) <event>> DelegateVoteChanged(delegate: address, previousBalance: uint, newBalance: uint) mint(\_to: address, \_amount: uint256) <<struct>> Checkpoint fromBlock: uint32 votes: uint256







# TimeLock.sol

# <<Library>> SafeMath

### Internal:

tryAdd(a: uint256, b: uint256): (bool, uint256)
trySub(a: uint256, b: uint256): (bool, uint256)
tryMul(a: uint256, b: uint256): (bool, uint256)
tryDiv(a: uint256, b: uint256): (bool, uint256)
tryMod(a: uint256, b: uint256): (bool, uint256)
add(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256): uint256
mul(a: uint256, b: uint256): uint256
div(a: uint256, b: uint256): uint256
mod(a: uint256, b: uint256): uint256
sub(a: uint256, b: uint256): uint256
mod(a: uint256, b: uint256, errorMessage: string): uint256
mod(a: uint256, b: uint256, errorMessage: string): uint256
mod(a: uint256, b: uint256, errorMessage: string): uint256

### Timelock

### Public:

GRACE\_PERIOD: uint MINIMUM\_DELAY: uint MAXIMUM\_DELAY: uint

admin: address

pendingAdmin: address

delay: uint

admin\_initialized: bool

queuedTransactions: mapping(bytes32=>bool)

### Internal:

getBlockTimestamp(): uint

External:

<<pre><<pre><<pre>payable>> null()

Public:

<<p>executeTransaction(target: address, value: uint, signature: string, data: bytes, eta: uint): bytes

<<event>> NewAdmin(newAdmin: address)

<<event>> NewPendingAdmin(newPendingAdmin: address)

<<event>> NewDelay(newDelay: uint)

<<event>>> CancelTransaction(txHash: bytes32, target: address, value: uint, signature: string, data: bytes, eta: uint)

<event>> ExecuteTransaction(txHash: bytes 32, target: address, value: uint, signature: string, data: bytes, eta: uint)

<event>> QueueTransaction(txHash: bytes32, target: address, value: uint, signature: string, data: bytes, eta: uint)

constructor(admin\_: address, delay\_: uint)

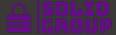
setDelay(delay\_: uint)

acceptAdmin()

setPendingAdmin(pendingAdmin\_: address)

queueTransaction(target: address, value: uint, signature: string, data: bytes, eta: uint): bytes32

cancelTransaction(target: address, value: uint, signature: string, data: bytes, eta: uint)



# BEP-20's Conformance

This test checks for BEP-20's conformance.

- All the functions are present
- All the events are present
- Functions return the correct type
- Functions that must be view are view
- Events' parameters are correctly indexed
- The functions emit the events
- Derived contracts do not break the conformance

Function	present	type	j	Correct	events	
				Return value		
totalSupply	<u>~</u>	V	view	<b>▽</b>		
balanceOf(address)	$\overline{A}$	$\overline{V}$	view	V		
transfer (address, uint 256)	$\overline{A}$	$\overline{\mathbf{V}}$	external	<b>✓</b>	<u> </u>	
					Transfer	
transferFrom(address,	<b>▽</b>	V	external	<b>▽</b>	<b>~</b>	
address, uint256)					Transfer	
approve(address,uint256)	<b>▽</b>	V	external	<b>✓</b>	<b>✓</b>	
					Approval	
allowance (address,	$\overline{\mathbf{V}}$	$\overline{\mathbf{V}}$	view	<b>✓</b>		
address)						
name	$\overline{A}$	$\overline{\mathbf{V}}$	view	<b>✓</b>		
symbol	<b>▽</b>	V	view	<b>▽</b>		

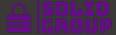
Check Events:



**▼** Transfer



Approve



# Findings

# LaboToken.sol

# <u>lssue #1:</u>

Type	Severity	Location
Gas Optimization	<ul> <li>Informational</li> </ul>	LaboToken.sol

# Description:

The public setStartBlock, add, set, deposit, withdraw, emergencyWithdraw, dev, setFeeAddress and updateEmissionRate functions should be declared as external.

# Recommendation:

These functions are only called from outside of the contract, consider using the external attribute instead of public to save gas.

# Summary

<ul> <li>Informational severity Issues</li> </ul>	1
<ul><li>Low severity issues</li></ul>	0
<ul><li>Medium severity issues</li></ul>	0
High severity issues	0



### MasterChefV2.sol

# Issue #1:

Туре	Severity	Location
Gas Optimization	<ul><li>Informational</li></ul>	Master Chef V2. sol

# **Description:**

The public setStartBlock, add, set, deposit, withdraw, emergencyWithdraw, dev, setFeeAddress and updateEmissionRate functions should be declared as external.

# Recommendation:

These functions are only called from outside of the contract, consider using the external attribute instead of public to save gas.

# Issue #2:

Туре	Severity	Location	
Best Practice	<ul> <li>Informational</li> </ul>	Master Chef V2. sol	

# Description:

Missing event in setStartingBlock function.

# Recommendation:

Consider adding event when setStartingBlock is execute.

```
function setStartBlock(uint256 _startBlock) public onlyOwner {
    require(startBlock == 0, "already started");
    require(_startBlock > block.number + 200, "start block has to be further in the future");
    startBlock = _startBlock;
    emit SetStartingBlock(startBlock);
}
```

# **Summary**

<ul> <li>Informational severity Issues</li> </ul>	2
<ul><li>Low severity issues</li></ul>	0
Medium severity issues	0
High severity issues	0