SOLID GROUP

Security Assessment

July 22nd, 2021

For:

Dexfolio Finance

Disclaimer

1. Solid Group assessment, audit and report below (the "**Reports**") are for informational purposes only and not, nor should be considered, as an endorsement to engage with, invest in, participate, provide an incentive, or disapprove, criticise, discourage, or purport to provide an opinion on any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any product, smart contract or asset, created by any team. It is further noted that Solid Group does not cover the testing, auditing or review of any integrated assets, such as external libraries, smart contracts or services (such as Unicrypt, Uniswap, SushiSwap, PancakeSwap etc.).

2. Solid Group do not provide any warranty or guarantee regarding the nature of the technology analysed, nor does Solid Group provide any indication of the ownership, or proprietary rights pertaining the technology analysed. Solid Group audits should not be used in any way to make decisions regarding investments or involvement with any particular coding project. These reports, in no way, provide investment advice, nor should be used as investment advice of any sort. As such, the use of these Reports is and shall be at all time at these report's recipient sole risk and the entire risk as to title, non-infringement, performance and accuracy is with the recipient of these reports.

3. By reading and/or making use of the Reports you are hereby deemed to have accepted, that **IN NO EVENT SHALL SOLID GROUP OR ANYONE ON ITS BEHALF BE LIABLE FOR ANY** (i) indirect, consequential, incidental, special or punitive damages of any kind, including without limitation damages for loss of business, savings or anticipated profits or revenues, business interruption, loss of business information, assets, or business opportunity, losses incurred as a result of, or in connection with, a mistaken or non-optimal analysis, or loss or damages to goodwill personal injury, property damages and/or monetary damages, of any nature whatsoever, including without limitation if arising out of the use of, or inability to use the Reports, regardless of the cause and whether arising in contract (including fundamental breach), tort (including negligence), or otherwise, even if the contractor has been advised of the possibility of such damages or loss or whether such losses or damages were otherwise foreseeable and as long as the above mentioned is not due to the contractor's fraud or wilful misconduct; (ii) errors, mistakes, inaccuracies, non-suitability, non-conformity or non-merchantability of any of the technologies reviewed as long as the above is not due to Solid Group's fraud or wilful misconduct; (iii) bugs, viruses, Trojan horses, or the like which may be transmitted by the technology reviewed hereunder, or any third party, nor any malfunctioning of a third party which could lead to the reviews being inaccurate.

With respect to token offering services, the recipient of these Reports is solely responsible to read the applicable white paper and all other documents concerning such token offering, and agree to the terms and conditions concerning stipulated in such third party token offering documents, the applicable token offering, and the decision to invest and participate in the applicable token offering vests solely with such recipient.

4. Accordingly, to the maximum extent permitted by applicable law, these Reports are provided "as is" and Solid Group and its affiliates hereby disclaim all warranties and conditions, either express, implied or statutory, including without limitation, any (if any) implied warranties or conditions of merchantability, profitability, fitness for a particular purpose, lack of viruses, title, non-infringement, quiet enjoyment or that the services will perform error-free or uninterrupted.

5. Solid Group is not liable under no circumstance, to any third party reliance on this Report or any information arising from it. This Report is not to be used, circulated, quoted or otherwise referred to for any purpose without our express prior written permission.

6. Solid Group Reports do not substitute any due diligence process and/or continuous security measure done by any company of individual.

7. Solid Group does not claim any guarantee of security or functionality of the reviewed Technology.

8. These Reports are rendered on the date hereof and Solid Group does not undertake, nor does it have any continuing obligation hereunder, to inform of any changes, updates, or other Reports conducted in connection with the contract presented and reviewed by Solid Group hereunder.

About Dexfolio

The only multi-DEX portfolio tracker with Automatic alerts. DEXF runs on BSC and is used for pro features, staking, and governance.

Website: http://dexfolio.org
Telegram: @dexfolioChat
Twitter: https://twitter.com/dexfolioapp
Medium: dexfolio.medium.com

About Solid Group

Solid Group is a blockchain consulting and auditing service provider, founded by 3 cybersecurity experts with a passion for thinking out of the box, learning, and sharing knowledge. Every project goes through a meticulous process and is viewed by at least two partners, thereby achieving a high level of credibility and professionalism. Our group is partnered with multiple organizations and launchpads that have a combined market cap of over 400 million USD.

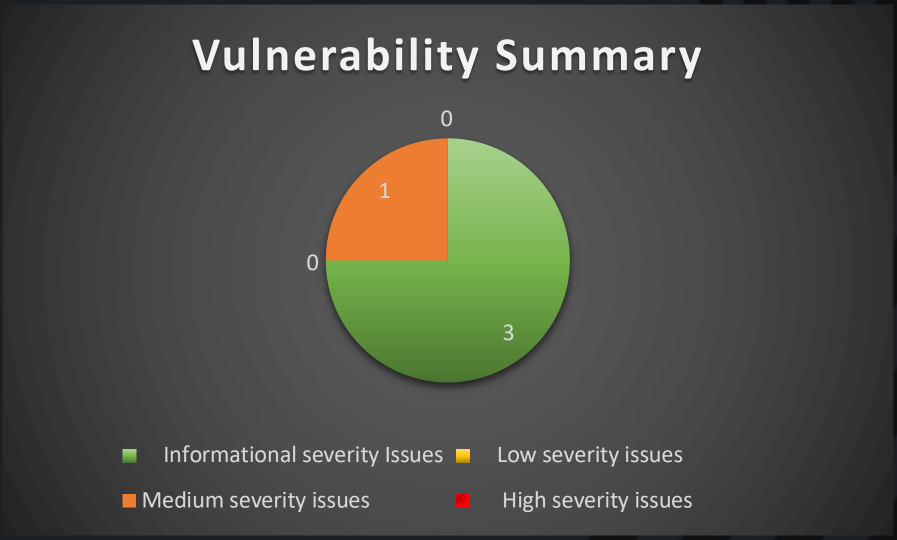
📣Telegram | 🗣Telegram discussion group |🐦 Twitter | 🛡 Contact for audit | 🤖 Audit Checker bot | Medium

Files in Scope

Contract Name	Contract Address (BSC)
LPFarming	0xe0AcBA75Dcd7C556a70201B7eaaf35B6d4B04C97
Timelock.sol	0x1F5b8300fed2EE70c3933C004C2B9e7089eC6aCb
DEXF.sol	0xB9844A9Cb6aBD9F86bb0B3aD159e37EeccE08987
GovernorAlpha .sol	0x0253E3Ad46c9Ca1df32c3d3EecC09f3f1A84ef74

Vulnerability Summary

● Informational severity Issues	3
● Low severity issues	0
● Medium severity issues	1
● High severity issues	0



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	Correct Return value	Events
totalSupply	☑	☑ view	☑	
balanceOf(address)	☑	☑ view	☑	
transfer(address,uint256)	☑	☑ external	☑	☑ Transfer
transferFrom(address, address, uint256)	☑	☑ external	☑	☑ Transfer
approve(address,uint256)	☑	☑ external	☑	☑ Approval
allowance(address, address)	☑	☑ view	☑	
name	☑	☑ view	☑	
symbol	☑	☑ view	☑	

Check Events:

☑ Transfer
☑ Approve

General:

☑ No external mint function
☑ No Volatile Code

Contract tested was the token’s contract: DEXF.sol

Tokenomics

- Owner – 40000000 tokens
- Treasury – 72000000 tokens
- Team – 20000000 tokens
- Staking Pool – 68000000 tokens
- DAILY_RELEASE_PERCENT_STAKING – 10% (can be changed)

DAILY_RELEASE_AMOUNT_TEAM – vested over 104 days

DAILY_RELEASE_AMOUNT_TREASURY – vested over 3647 days

Dexf.sol

Privileged Functions

- setDailyReleaseAmountTreasury - The owner of the contract can change the number of tokens that are being released every day.
- setDailyReleasePercentStaking **-** The owner of the contract can change the percentage of tokens that are being released every day to the staking contract.
- setStakingContract – This function can only be called once. This function sets the stakingContract address.
- setStakingRewardRemaining – The owner can set the remaining tokens for staking rewards.
- setTreasury1 – The owner can set the treasury address to any address desired.
- setTaxFee – The owner can set the % that will go to the staking contract, the tax fee cannot be greater than 10%.
- addToBlacklist – The owner can blacklist any address within the BLACK_AVAILABLE_PERIOD excluding the pair address which could cause the token to be untradable.
- The token can be paused by the owner.
- setEpoch1Start - The owner can set the staking start time and affect the rewards. This happens due to the fact that the rewards are calculated from this time. In addition, the owner can reset the start time.
- changeAllocation- This function gives the owner of the contract the option to dynamically change the token allocation between _team, _stakingPool, _treasury1.
- removeFromBlacklist – The owner can remove an address from blacklist within the following timeline BLACK_AVAILABLE_PERIOD.
- updateBuyLimit – The owner can set the maximum amount of tokens that can be bought in one transaction. The minimum value is 7000 tokens.
- updateSellLimit – The owner can set the maximum amount of tokens that can be sold in one transaction. The minimum value is 7000 tokens.

This token is a pausable token which means the owner can pause _transfer any time.

Dexf.sol Findings

Issue #1:

Type	Severity	Location	Status
Logical Issue	● Informational	_isBuy, _isSell	✅ Acknowledged

Description

There is no way to differentiate between sell transaction and addLiquidity transaction and buy transaction and removeLiquidity transaction by just looking at the sender/recipient.

Therefore, _isBuy will also return True for removeLiquidity transaction, and _isSell will also return true for addLiquidity transactions.

Issue #2

Type	Severity	Location	Status
Logical Issue	● High	buyLimit & sellLimit	✅Fixed

Description

In your current code, when you first initialize the market you can't set any limits if the amount of tokens that is planned for the first market initialization is greater than sellLimit, Because there is no way to differentiate between liquidity addition and sell transaction.

Most of the bots usually buy on the same blocks or few blocks after. (Related to issue #1)

Recommendation

Exclude the owner or the address which initializing the market from the selling limitation. Make sure the exclusion will be only once (when initializing the market).

Issue #3

Type	Severity	Location	Status
Owner Capabilities	● High	buyLimit & sellLimit	✅Fixed

Description

addToBlacklist function was added to prevent bots on the listing, the owner can blacklist addresses indefinitely and prevent certain addresses from buying/selling. The blacklist function shouldn’t be used after the first minutes from listing.

Recommendation

Our recommendation is to limit the timeframe the owner has to append an address to the blacklist.

Issue #4

Type	Severity	Location	Status
Owner Capabilities	🔴 High	buyLimit & sellLimit	✅ Fixed

Description
During the blacklist timeframe period the owner can blacklist address he could also blacklist the pair address which will make the token untradable since each transfer consist the pair address.

Recommendation

Our recommendation is to prevent the owner from blacklisting the pair address, and the router address.

The team added a require statement that would prevent the pair from being blacklisted. The router can still be blacklist and effect the addLiquidity / removeLiquidity functionality.

Issue #5

Type	Severity	Location	Status
Best Practice	🟢 Informational	All	✅ Fixed

Description
Lack of events in the contract. Events should be added to all the functions that change important variables and contract functionality.

- removeFromBlacklist
- updateBuyLimit
- setPairAddress
- updateSellLimit
- withdrawFromTreasury
- claimStakingReward
- _initEpoch

Issue #6

Type	Severity	Location	Status
Best Practice	● Informational	setPair	✅ Fixed

Description

The pair address is set manually.

Recommendation

The pair address could be automatically calculated following this documentation, no need to set the pair address manually. If the pair address won't be properly set before listing the buy and sell limitation won't work.

Issue #7

Type	Severity	Location	Status
Owner Capabilities	● High	withdrawFromTreasury	✅ Fixed

Description

The owner of the contract can withdraw all the tokens that were allocated for the treasury.

Recommendation

Remove this function if not needed.

Issue #8

Type	Severity	Location	Status
Owner Capabilities	● High	withdrawFromTreasury	✅ Fixed

Description

The owner of the contract can call claimStakingReward with any amount he wants and withdraw any amount of tokens from the _stakingPool.

Recommendation

Depends on the project's design. One possibility is to limit the set function to occur only once or only when the contract is protected by a timelock contract.

Issue #9

Type	Severity	Location	Status
Owner Capabilities	● High	setDailyReleasePercentStaking	✅ Fixed

Description

The team can control the number of tokens that are released everyday by calling setDailyReleasePercentStaking. The owner can bypass the vesting limitation and release the whole amount in a specific day by setting the daily percentage to 100%. Then, the owner can call changeAllocation function and transfer all the tokens which were allocated for the staking rewards to the team.

Recommendation

Related to issue 10

Issue #10

Type	Severity	Location	Status
Owner Capabilities	● High	changeAllocation	✅ Fixed

Description

This function gives the owner of the contract the option to dynamically change the token allocation between _team, _stakingPool, _treasury1. As part of this function, the owner can transfer any amount of tokens to the _team address from the tokens which are allocated for stakingPool and treasury pool. **The owner can take any amount of tokens from these pools regardless to the amount of tokens which are unlocked.**

Recommendation

Our recommendation is to exclude the team address from getting tokens which were allocated for the staking pool and the treasury.

Issue #11

Type	Severity	Location	Status
Owner Capabilities	● High	setTaxFee, updateSellLimit, updateBuyLimit	✅ Fixed

Description

The owner of the contract can make the tokens untradable. By calling updateBuyLimit(0) or updateSellLimit(0) or by setting _taxFeeto a significant %. (Pancakeswap won’t work if the fees are higher than a certain value) or by adding the pair address to blacklist.

Recommendation

Our recommendation is to have a minimum or at least maximum limit for the following setter functions: setTaxFeePercent. Regarding the second issue add a require statement that would limit setting buyLimit or sellLimit to 0.

Regarding the third issue our recommendation is to prevent blacklisting the pair address in the code.

LPFarming.sol

Privileged Functions

- setMultipliers - The owner can choose the multiplier for each lock period.
- setEpoch1Start – Sets the time the staking contract is activate and rewards are received (can be re-set)

LPFarming.sol Findings

Issue #1

Type	Severity	Location	Status
Gas Optimization	● Informational	Constructor	✅ Fixed

Description

If the pair was not created by calling createPair, the getPair function will return address(0).

Recommendation

Our recommendation is to calculate the pair address regardless of the creation of the pair by following this documentation.

Issue #2

Type	Severity	Location	Status
Best Practice	● Informational	Constructor	✅ Fixed

_epoch1Start = block.timestamp + 3600 * 24 * 7 * 6;

_epochDuration = 86400;

Recommendation

Solidity supports weeks keyward consider changing the code for readability:

_epoch1Start = block.timestamp + 6 weeks;

_epochDuration = 24 hours;

Issue #3

Type	Severity	Location	Status
Best Practice	● Informational	swapBNBForTokens	✗ Not Fixed

Description

swapBNBForTokens calls external function swapExactETHForTokensSupportingFeeOnTransferTokens it is customary when calling to external function to use try-catch.

Recommendation

Use try-catch when calling external function such as swapExactETHForTokensSupportingFeeOnTransferTokens.

Issue #4

Type	Severity	Location	Status
Best Practice	● Informational	swapBNBForTokens	✗ Not Fixed

Description

swapBNBForTokens calls external function swapExactETHForTokensSupportingFeeOnTransferTokens it is customary when calling to external function to use try-catch.

Recommendation

Use try-catch when calling external function such as swapExactETHForTokensSupportingFeeOnTransferTokens.

Issue #5

Type	Severity	Location	Status
Best Practice	● Informational	addLiquidityBNB	✗ Not Fixed

Description

addLiquidityBNB calls external function addLiquidityETH it is customary when calling to external function to use try-catch.

Recommendation

Use try-catch when calling external functions such as addLiquidityETH.

Issue #6

Type	Severity	Location	Status
Logical Issue	● High	Stake	✅ FIxed

Description

The swapAndLiquifyFromBNB function converts half of the BNB to tokens. The other half of BNB and part of the converted BNB to tokens are deposited into the DEXF-BNB pool on pancakeswap as liquidity.

Every time the swapAndLiquify function is called, a small number of tokens are leftover in the contract instead of refunding the user for the nonoptimal ratio, because of the price of the token increases after swapping the first half of BNB into tokens. Therefore, the other half of BNB tokens require fewer tokens than the converted tokens to be paired with it when adding liquidity. The contract doesn't appear to provide a way to refund the user for the missing tokens.

Recommendation

Our recommendation is to refund the leftovers tokens/BNB to the staker.

Issue #7

Type	Severity	Location	Status
Best Practice	● High	All	✅ FIxed

Description

Add an emergency unstake function that would only withdraw their funds without calling any external function / any function that would likely fail in order to ensure investors' funds are safe.

Issue #8

Type	Severity	Location	Status
Logical Issue	● High	stakeDexf , stakeToken	✅ Fixed

Description

The code will malfunction if the staked token is a token with fees on transfer.
The problem is that if there is a transfer fee, the amount of tokens staked will be less than tokenAmount, while the amount of tokens that is being swapped for liquidity is equal to tokenAmount which is less than the amount received from transferFrom (due to fees).

Recommendation

Our recommendation is to save the initial balance before the transfer and get the balance after the transfer, The difference will be the actual balance after deducting the fees.

Issue #9

Type	Severity	Location	Status
Gas Optimization	● Informational	safeDexfTransfer	✅ Fixed

Description

Unused internal function in the code.

Recommendation

Remove internal function from the code to save on storage.

Issue #10

Type	Severity	Location	Status
Volatile Code	● Potentially High	getClaimableAmountByIndex	❌ Not Fixed

Description

The code consumes large amount of gas, and can potentially reach block gas limit which may cause the withdraw function to be uncallable. The team has added an emergency function that should always work in case there is an issue with the contract.

Recommendation

Our recommendation is to minimize the maximum staking time period so the gas consumes by iterating the days will be less and reasonable.

Issue #11

Type	Severity	Location	Status
Logical Issue	● Medium	getClaimableAmountByIndex	✗ Not Fixed

Description

If _dailyStakingRewards of a certain day were 0, all users that staked during that day will not get rewards for the rest of their staking period.
Note that _dailyStakingRewards can be set to 0 using privileged functions such as changeAllocation, setStakingRewardRemaining and setDailyReleasePercentStaking.