

SOLID
GROUP

X

LOVAR

Security Assessment

June 29th, 2021

For:
Lunar Token

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. Solid group do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc’...)

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 technology 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.




About LunarSwap

LunarSwap is an innovative project, all of your Lunar tokens will be automatically staked for you to be rewarded in BNB and LunarSwap tokens

[Website](#) | [Twitter](#) | [Telegram](#)

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
Initial Revision	https://github.com/lunar-token/lunartoken/commit/0896e60a652f293a4a774719cb44fcc25dd16abe
Second Revision	https://github.com/lunar-token/lunartoken/commit/6d9fc824d61ea691e93845e6660e3dd524a5d746

Vulnerability Summary

● Informational severity Issues	0
● 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: **Official Mars Token**

Issue #1	Volatile code	● High	_tokenTransfer	✓ Fixed
----------	---------------	--------	----------------	---------

Description

Errors should be handle gracefully especially in critical functions such as `_transfer`. `_transfer` calls `dividendsOf`.
<https://github.com/lunar-token/lunartoken/blob/0896e60a652f293a4a774719cb44fcc25dd16abe/Lunar.sol#L192>

Recommendation

If there is an overflow error, the token won't be tradable. Make sure that even if there is an error like that., the token is tradable by not using require.

Issue #2	Logical Issue	● Medium	_transfer	✓ Fixed
----------	---------------	----------	-----------	---------

Description

```
if (!_isExcludedFromFee[sender] || !_isExcludedFromFee[recipient]) {
```

If the contract will not be excluded from fee, it will cause a recursive call when calling swap, since `swapTokensForBnb` calls `_transfer` internally.

Recommendation

Add a modifier that would prevent the recursion.

Fix

The team need to make sure the contract is **excludedFromFee**.

Issue #3	Logical Issue	● High	_transfer	✓ Fixed
----------	---------------	--------	-----------	---------

Description

When an address is excluded from staking, their `stakeValue` entry is reset. This mean their rewards will be reset, therefore they won't get their rewards and they will be stuck in the contract.

Recommendation

Our recommendation is to send the rewards to the user when calling `excludeFromStaking`.

Issue #4	Logical Issue	● High	_transfer	✓ Fixed
----------	---------------	--------	-----------	---------

Description

_transfer should always work, even if there is a bug in the contract, to ensure that investors' funds are safe. If the function is critical (such as _transfer) always make sure its error cases are handled gracefully! _transfer calls _checkSwapViability which could fail.

Recommendation

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

Issue #5	Logical Issue	● High	includeInStaking	✓ Fixed
----------	---------------	--------	------------------	---------

Description

includeInStaking calculates the stakeValue variable based on the user's _tOwned, which was calculated in excludeFromStaking based on the user's _rOwned. If the user is excluded and then included the stakeValue of the user includes the RFI rewards the user received before he was initially excluded. This means the user will receive staking rewards on the RFI rewards.

Recommendation

To fix this issue the original stakeValue variable must be saved instead of being zeroed out in excludeFromStaking.

Issue #6	Logical Issue	● High	restake	✓ Fixed
----------	---------------	--------	---------	---------

Description

restake doesn't update the totalStaked value.

Issue #7	Logical Issue	● Medium	restake	✓ Acknowledge
----------	---------------	----------	---------	---------------

Description

The code does not differentiate between token sells and liquidity addition, since both of them send tokens to the pair. Similarly, token buys and liquidity removal will also look the same since they send tokens from the pair

⚠ swapAndDistribute takes place only on sell transactions. Therefore the contract will accrue a significant amount of tokens. This may result in a pretty significant price drop because of the large number of tokens that would enter the pool. The team should be aware of this issue.