SOLID GROUP

X

Security Assessment

July 27th 2021

For:
Football Fantasy

SOLID GROUP

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.

SOLID GROUP

About Football Fantasy

Fantasy Football evolves in the crypto world! Build your very own Football team in crypto and compete on tournaments to win daily, weekly, monthly prizes and much more!
🌍 https://footballfantasypro.com
https://twitter.com/Footballstarsio
https://www.instagram.com/footballstarsio/

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: https://t.me/solidgroup

Contact For an audit: https://t.me/solid_1

Twitter: https://twitter.com/solid_group_1

Medium: https://solidgroup-68170.medium.com/

Files In Scope

Contract Name	Contract Address
FootballFantasy	0xcAC33Ce2734D30949F5a96f7d64851830fDa7AD9

Vulnerability Summary

Informational severity Issues	2
Low severity issues	1
Medium severity issues	3
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

Findings

Issue #1:

Type	Severity	Location
Logical Issue	Informational	`includeInReward`

Description

The error message doesn't match the condition.

```
require(_isExcluded[account], "Account is already excluded");
```

Issue #2:

Type	Severity	Location
Lack of events	Medium	

Description

The contract uses a modified version of Ownable contract. These modifications have a significant flaw — a malicious owner can get his owner capabilities even after calling renounceOwnership.

Recommendation

Remove lock and unlock functions.

Issue #3:

Type	Severity	Location
Lack of events	Medium	`includeInReward`

Description

The code is vulnerable to the SafeMoon bug - excluding an address from the fee and then later including it back will cause the address to receive all RFI rewards for the time it was excluded (at the expense of other holders).

In includeInReward _rOwned is not updated. _rOwned should be updated and be calculated according to the current rate.

Recommendation

Properly calculate _rOwned of the included address in includeInReward based on its _tOwned amount

Issue #4:

Type	Severity	Location
Gas Optimization	Informational	expectedRewards tokenHolder numberOfTokenHolders exist myRewards

Description

Unused code. it is recommended to clean unused code before deployment to save gas fees and storage.

Issue #5:

Type	Severity	Location
Volatile Code	Low	_transfer

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 swapTokensForEth and addLiquidity which could fail when calling swapExactTokensForETHSupportingFeeOnTransferTokens and addLiquidityETH.

Recommendation

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

Issue #6:

Type	Severity	Location
Centralization Issue	● Medium	addLiquidityETH

Description

The recipient of the newly created LP tokens is the owner of the contract. The newly created LP tokens are unlocked.

Recommendation

Our recommendation is to change the recipient of the newly created LP tokens to the contract in order to ensure that the LP tokens are locked or to simply lock the tokens in the contract for a certain period.

Issue 7:

Type	Severity	Location
Logical Issue	● Low	addLiquidityETH

Description

SwapAndLiquify uses 2 quarters of the contract's token balance for liquidity addition – 1 quarter of the tokens are paired with another quarter that is converted to BNB.
However, since the price of the token drops after executing the first conversion, this may cause leftover BNB to get stuck in the contract.

Recommendation

Our recommendation is to use the leftover BNBs for buyback.