SOLID GROUP

X

Security Assessment

July 14th , 2021

For:
Wizard

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 Wizard

Wizard is Fantasy-based NFT platform which will have other fantasy-themed tokens in the future. Through having great and flexible utility both in-game, and in staking, we will have a lot to offer everyone. Having NFTs integrated into our games unlocks a vast amount of potential, we can let NFT holders have access to new levels, characters, spells, etc. And we can allow players to win NFTs from beating our games, and combine those NFTs with others to create even more rare and valuable ones that can be used for further staking and gameplay.

🌏 https://wizard.financial/

🐦 https://twitter.com/WIZARD_BSC

https://t.me/wizard_financial

Medium: https://wizardtokenofficial.medium.com/

Discord: https://discord.com/invite/dfKrgACzHx

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.

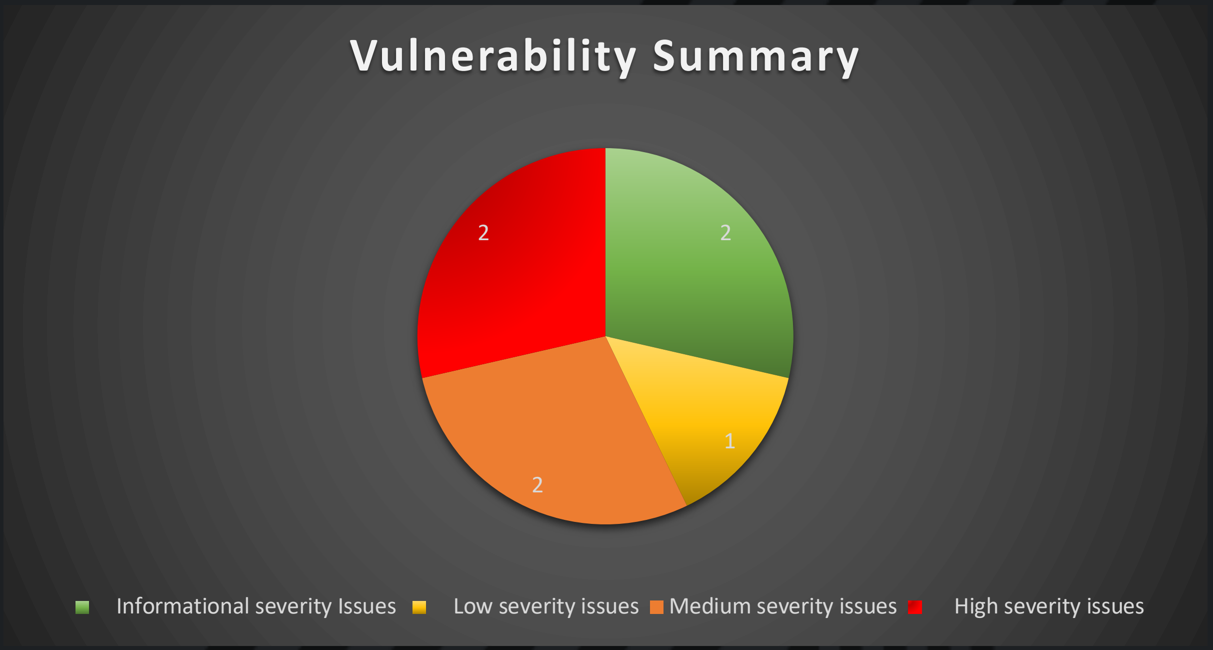
📣 Telegram | 🗣Telegram discussion group |🐦 Twitter | 🛡 Contact for audit | 🤖 Audit Checker bot | Medium

Files in Scope

Contract Name	MD5
MD5	4ea40b532acd0 4ea40b532acd0f637984972887771764 f637984972887771764

Vulnerability Summary

● Informational severity Issues	2
● Low severity issues	1
● Medium severity issues	2
● High severity issues	2



Privilege Functions

- The owner of the contract can blacklist any address, note that blacklisted addresses cannot sell nor buy tokens. The owner of the contract could also blacklist the pair address which will make the token untradable.

- The owner of the contract can change the number of tokens the contract will sell in one transaction by calling changeminTokenNumberToSell.
- The owner can call changecharitywallet
- The owner of the contract can set all the fees as long as they are less or equal to 5%.
- The recipient of the newly created LP tokens is the owner of the contract. The newly created LP tokens are unlocked.

Findings

Issue #1	Type	Severity	Location	Status
	Owner Capabilities	● High	setTaxFeePercent setLiquidityFeePercent setBurnFeePercent setCharityFeePercent setMaxTxPercent	✗ Not Fixed

Description

The owner of the contract can make the tokens untradable. By calling setMaxTxPercent(0) or by setting _taxFee variable or _liquidityFee, or _charityFee variables to a significant %. (Pancakeswap won't work if the fees are bigger than a certain value)

Recommendation

Our recommendation is to have a minimum or at least maximum limit for the following setter functions: setTaxFeePercent, setLiquidityFeePercent , setBurnFeePercent , setCharityFeePercent.
Add a required statement that would limit setting _maxTxAmount to 0.

Issue #2	Type	Severity	Location	Status
	Owner Capabilities	● High	addBlacklist	✗ Not Fixed

Description

addBlacklist function was added to prevent bots on the listing, the owner could still blacklist addresses forever and prevent certain addresses from buying/selling. The blacklist function shouldn't be used after listing

Recommendation

Our recommendation is to limit the timeline window when the owner can append an address to the blacklist.

Issue #3	Type	Severity	Location	Status
	Gas Optimization	● Informational	_transferStandart _transferToExcluded _transferFromExcluded	✗ Not Fixed

Description

If the charity address is not excluded, there is no need to update _tOwned.

```
tOwned[charityaddress] = _tOwned[charityaddress].add(tcharityFee);
_rOwned[charityaddress] = _rOwned[charityaddress].add(rcharityfee);
```

Issue #4	Type	Severity	Location	Status
	Best Practice	● Informational		✗ Not Fixed

Description

Lack of events in the contract.

Recommendation

Our recommendation to emit events when changing the state variable of the contract

Issue #5	Type	Severity	Location	Status
	Logical Error	● High	_transfer	✅ Fixed

Description

A logical error in this condition. In order to blacklist anyone from selling/buying uniswap pair address will have to be blacklisted as well.

```
require(blacklist[from] == false && blacklist[to] == false, "Blacklist address found");
```

Recommendation

```
require(blacklist[from] == false || blacklist[to] == false, "Blacklist address
```

Issue #6	Type	Severity	Location	Status
	Volatile code	● Medium	_transfer	✗ Not Fixed

Description

_transfer function should always work, even if a bug was found in the contract. In order to ensure that investors' funds are safe & secured.
If a function is **mandatory (such as _transfer) our state of mind is to 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 #7	Type	Severity	Location	Status
	Owner Capabilities Issue	● Medium	addLiquidity	✗ Not Fixed

Description

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

```
uniswapV2Router.addLiquidityETH{value: ethAmount}(
    address(this),
    tokenAmount,
    0, // slippage is unavoidable
    0, // slippage is unavoidable
    owner(),
    block.timestamp
  );
```

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 locked the tokens in the contract for a certain time frame.

Issue #8	Type	Severity	Location	Status

	Logical Issue	● Low	lock()
			unlock()
				✗ Not Fixed

Description

An owner has the ability to gain ownership of the contract even if he calls renounceOwnership function.

This can be achieved by performing the following steps:

1. The owner of the contract can call lock() function to lock the contract (the lock function saves the previous owner into a variable)
2. After the locking period has passed the owner of the contract can call unlock() and regain the ownership.
3. The owner of the contract can then call the renounceOwnership function. Now, the contract allegedly has no owner (users can verify it by looking for the renounceOwnership transaction and making sure that the owner is set to the zero address)
4. The owner of the contract can call the unlock function again, and get the ownership back.

Recommendation:

Our recommendation Is to remove the lock and unlock function if not needed.