AUDIT REPORT

23 MAY 2021

FOR

SUPERSHIBA.EXCHANGE

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. 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 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: Binance Smart Chain
Website: SuperShiba.xyz
Twitter: Twitter.com/SuperShibabsc
Telegram Group: T.me/SuperShibabsc

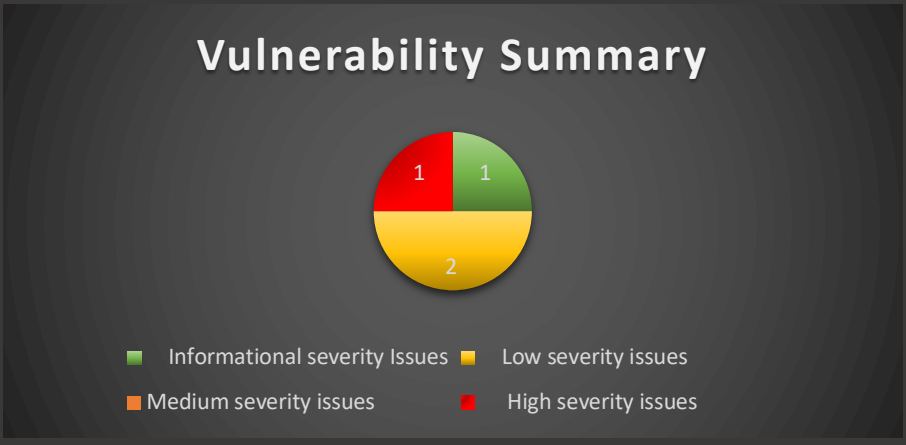
Description

The Super Shiba DEX is something revolutionary that the BSC has been in need of- this exchange will only featured fully audited and KYC'd tokens so that team has approved to list on the exchange. This means when a token is listed on the exchange you can know it's a safe investment. The team are fully DOXXED, and the roadmap is full of amazing moves. The platform will have lucrative farms and pools, a lottery and even a presale platform.

Files in Scope

Contract Name	Contracts (BSC)
treats.sol	0x570919a72db73799191fc734b6D600750C0Dc18C
masterchef.sol	Not deployed yet
factory.sol	0x6d3703334eCbB0a74E0A67Fe5040a9d850cEcF0E
router.sol	0x8Ba22ebAFcBC717086f94d13dBa5a5F1d06F9DaC
Timelock.sol	Not deployed yet

Vulnerability Summary



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:

⚠ The contract has a mint function; Solid Group will confirm that the ownership was transferred to the masterchef contract after deployment.

✅ No Volatile Code

The contract that was tested is the token's contract: treats.sol

Findings

Issue #1:

Type	Severity	Location	Status
Owner Capabilities	High	MasterChef.sol	Fixed ✅

Description:

The contract takes deposit fee when staking is applied, however there is no limitation for the deposit fee value.

This might cause the owner of the contract to set the _depositFeeBP variable to 100% and take 100% of the new user staking.

Recommendation:

Our recommendation is to set a high limit for_depositFee variable via add() and set() functions.

Fix:

✅ deposit fee can't be greater than 10%.

```
require(_depositFeeBP <= 1000, "add: invalid deposit fee basis points");
```

Issue #2

Type	Severity	Location	Status
Volatile Code	Low	MasterChef.sol	Not Fixed ❌

Description:

The implementation of _lpToken parameter and safeTransfer() function is unknown, both of them are being used widely across contract.

That being said, they may have a malicious logical implementation that calls to the function deposit(), and as a result could lead to another invocation of safeTransfer() without updating user.amount variable.

This could jeopardize user.amount and it will cause a miscalculation in user's balance eventually.

Issue #3:

Type	Severity	Location	Status
Best Practice	Low	MasterChef.sol	Not Fixed ❌

Description:

Return values of external functions calls are being ignored and not reviewed. i.e. the return value of transfer() function transfer is being ignored via safeEggTransfer() function.

```
function safeEggTransfer(address _to, uint256 _amount) internal {
    uint256 eggBal = egg.balanceOf(address(this));
    if (_amount > eggBal) {
        egg.transfer(_to, eggBal);
    } else {
        egg.transfer(_to, _amount);
    }
}
```

Recommendation:

Our recommendation is to always check the return value when calling external functions.

Issue #4:

Type	Severity	Location	Status
Lack of Events	Informational	MasterChef.sol	Not fixed ❌

Description:

Missing events for critical operations. Note that events are highly important for the integration with certain DApps in the future. Functions that we have inspected that suffer from lack of events:

- Dev
- setFeeAddress()
- updateEmissionRate()

Recommendation:

Our recommendation is to always emit events.
i.e.
-when the core state of the contract is changed
-when changing the emission rate
-changing dev address
-change fee address
Etc.

Issue #5:

Type	Severity	Location	Status
Centralization	High	factory.sol	Not fixed ❌

Description:

Only feeToSetter()'s address has permissions to create new pairs in the system.
Mishandling a private key could cause devastated consequences on the project as a whole.

Recommendation:

Our recommendation is to have a list of authorized users in order to handle these kind of tasks. If one address crashes you will have additional backups that can support the contract.