

AUDIT REPORT

DATE APRIL 23RD

FOR

COSHI INU PROJECT





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: Ethereum
 Website: <https://corgishiba.dog>
 Twitter: <https://twitter.com/shibacorgidog>
 Telegram Group: <https://t.me/corgishibadog>

Description

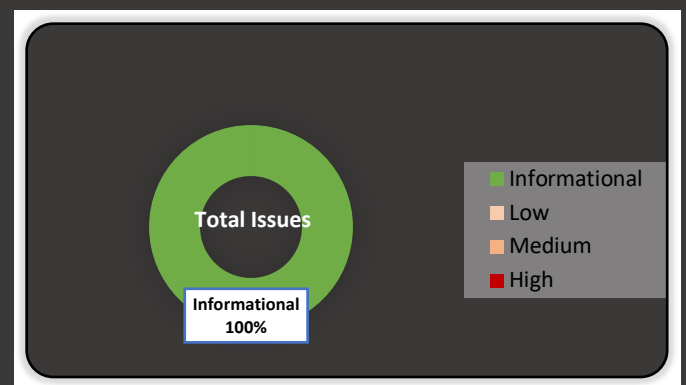
CoShi Inu is a MEME token on Ethereum blockchain. CoShi Inu is ETH version of ShiCo (@ShibaCorgiDog) which operates on Binance Smart Chain. Inspired by Shiba Inu.

Files in Scope

Contract Name	Contract (ETH)
CoShi Inu (CoShi).sol	0x668C50B1c7f46EFFBE3f242687071d7908AAB00A

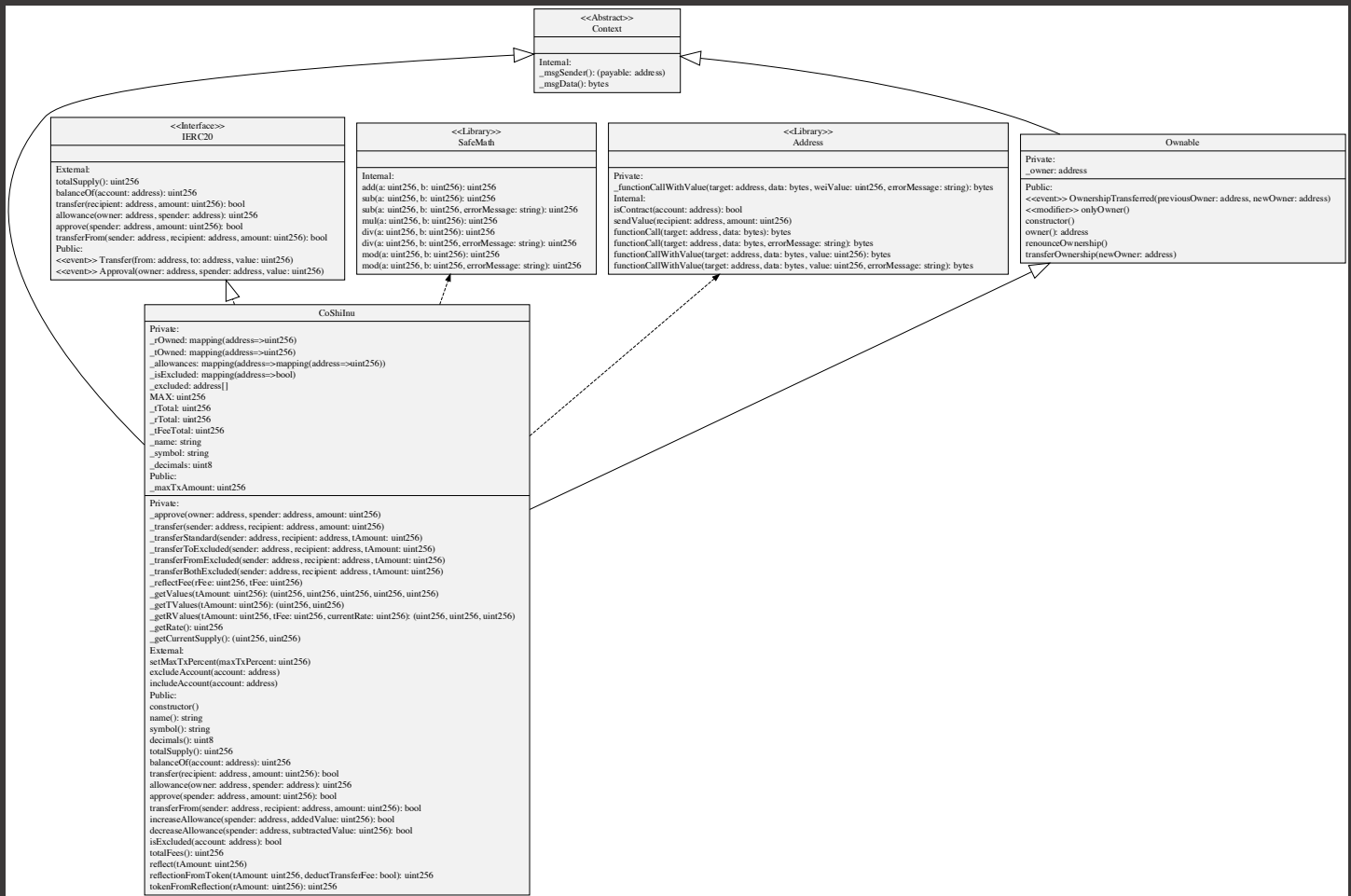
Vulnerability Summary

● Informational severity Issues	4
● Low severity issues	0
● Medium severity issues	0
● High severity issues	0



UML

CoShi Inu (CoShi).sol



ERC-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

The contract that was tested is the token's contract: CoShi Inu (CoShi)

Findings

Issue #1:

Type	Severity	Location
Gas Optimization	● Informational	CoShi Inu (CoShi).sol

Description:

The public `isExcluded`, `totalFees`, `reflect` and `reflectionFromToken` functions should be declared as external.

Recommendation:

These functions are only called from outside of the contract, consider using the external attribute instead of public to save gas.

Issue #2:

Type	Severity	Location
Gas Optimization	● Informational	CoShi Inu (CoShi).sol

Description:

State variables that are never changed should be constant.

Recommendation:

The state variable `“_name”`, `“_symbol”` and `“_decimals”` are never changed. Consider changing their attribute to constant to save on gas.

Issue #3:

Type	Severity	Location
Bast Practice	● Informational	CoShi Inu (CoShi).sol

Description:

`_tTotal` value has too many digits. Literals with too many digits are difficult to read and review.

Recommendation:

Our recommendation is to represent the number by a power of 10.

Issue #4:

Type	Severity	Location
Gas Optimization	● Informational	CoShi Inu (CoShi).sol

Description:

There are many places in the code that interact with `taxFee` variable, but since it is set to zero it has no effect. Therefore, the user pays extra gas costs for no reason.

Recommendation:

Our recommendation is to delete code that has no effect.

Summary

● Informational severity Issues	4
● Low severity issues	0
● Medium severity issues	0
● High severity issues	0

General Notes

✔ Ownership was renounced

<https://etherscan.io/tx/0x2b7ec3ecf910a95c49b82f3b94a7723942ad6a8fa9da482fcefa0d7029813c0d>