

REPORT 647EA1D008F98E0019B4979A

Created	Tue Jun 06 2023 03:02:40 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	6479a178a86727385335715d

## REPORT SUMMARY

Analyses ID	Main source file	Detected vulnerabilities
<a href="#">017aeb93-92e9-44e4-8c47-4e2aef04ad49</a>	gougou/gg.sol	7

Started	Tue Jun 06 2023 03:02:48 GMT+0000 (Coordinated Universal Time)
Finished	Tue Jun 06 2023 03:03:47 GMT+0000 (Coordinated Universal Time)
Mode	Deep
Client Tool	Remythx
Main Source File	Gougou/Gg.sol

DETECTED VULNERABILITIES

HIGH	MEDIUM	LOW
0	0	7

ISSUES

UNKNOWN Arithmetic operation "-" discovered  
This plugin produces issues to support false positive discovery within MythX.  
SWC-101

Source file  
gougou/gg.sol  
Locations

```
335 | require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
336 | unchecked {
337 |   _approve(sender, _msgSender(), currentAllowance - amount);
338 | }
```

UNKNOWN Arithmetic operation "+" discovered  
This plugin produces issues to support false positive discovery within MythX.  
SWC-101

Source file  
gougou/gg.sol  
Locations

```
354 | */
355 | function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) {
356 |   _approve(_msgSender(), spender, allowances[_msgSender()][spender] + addedValue);
357 |   return true;
358 | }
```

## UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
376 | require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below zero");
377 | unchecked {
378 |   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
379 | }
```

## UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
409 | require(senderBalance >= amount, "ERC20: transfer amount exceeds balance");
410 | unchecked {
411 |   _balances[sender] = senderBalance - amount;
412 | }
413 | _balances[recipient] += amount;
```

## UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
411 | _balances[sender] = senderBalance - amount;
412 | }
413 | _balances[recipient] += amount;
414 |
415 | emit Transfer(sender, recipient, amount);
```

## UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
432 | _beforeTokenTransfer(address(0), account, amount);
433 |
434 | totalSupply += amount;
435 | _balances[account] += amount;
436 | emit Transfer(address(0), account, amount);
```

## UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
433 |
434 | _totalSupply += amount;
435 | _balances[account] += amount;
436 | emit Transfer(address(0), account, amount);
```

## UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
458 | require(accountBalance >= amount, "ERC20: burn amount exceeds balance");
459 | unchecked {
460 | _balances[account] = accountBalance - amount;
461 | }
462 | _totalSupply -= amount;
```

## UNKNOWN Arithmetic operation "-=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
460 | _balances[account] -= accountBalance - amount;  
461 | }  
462 | totalSupply -= amount;  
463 |  
464 | emit Transfer(account, address(0), amount);
```

## UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
567 | manage = _manage;  
568 |  
569 | uint256 amount = _totalSupply / 10;  
570 | _mint(airdrop, amount);  
571 | _mint(communityRewards, 3 * amount);
```

## UNKNOWN Arithmetic operation "\*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
569 | uint256 amount = _totalSupply / 10;  
570 | _mint(airdrop, amount);  
571 | _mint(communityRewards, 5 * amount);  
572 | _mint(liquiditySupply, 5 * amount);  
573 | _mint(address(this), amount / 2);
```

## UNKNOWN Arithmetic operation "\*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
570 | _mint(airdrop, amount);
571 | _mint(communityRewards, 3 * amount);
572 | _mint(liquiditySupply, 5 * amount);
573 | _mint(address(this), amount / 2);
574 | _mint(manage, amount / 2);
```

## UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
571 | _mint(communityRewards, 3 * amount);
572 | _mint(liquiditySupply, 5 * amount);
573 | _mint(address(this), amount / 2);
574 | _mint(manage, amount / 2);
575 | }
```

## UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
572 | _mint(liquiditySupply, 5 * amount);
573 | _mint(address(this), amount / 2);
574 | _mint(manage, amount / 2);
575 | }
```

## UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
609 |
610 | if (limited && from == uniswapV2Pair) {
611 |     require(super.balanceOf(to) + amount <= maxHoldingAmount && super.balanceOf(to) + amount >= minHoldingAmount, "Forbid");
612 | }
613 | }
```

## UNKNOWN Arithmetic operation "+" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
609 |
610 | if (limited && from == uniswapV2Pair) {
611 |     require(super.balanceOf(to) + amount <= maxHoldingAmount && super.balanceOf(to) + amount >= minHoldingAmount, "Forbid");
612 | }
613 | }
```

## UNKNOWN Arithmetic operation "/" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
623 | _beforeTokenTransfer(sender, recipient, amount);
624 |
625 | uint256 donateAmount = amount * percentage / 1000;
626 | if (sender == donator || sender == communityRewards || sender == airdrop || sender == manage || sender == address(this) || recipient == uniswapV2Pair) {
627 |     donateAmount = 0;
```

## UNKNOWN Arithmetic operation "\*" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
623 | _beforeTokenTransfer(sender, recipient, amount);
624 |
625 | uint256 donateAmount = amount * percentage / 1000;
626 | if (sender == donator || sender == communityRewards || sender == airdrop || sender == manage || sender == address(this) || recipient == uniswapV2Pair) {
627 |     donateAmount = 0;
```

## UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
627 | donateAmount = 0;
628 | }
629 | uint256 transferAmount = amount - donateAmount;
630 | uint256 senderBalance = _balances[sender];
631 | require(senderBalance >= amount, "ERC20: transfer amount exceeds balance");
```

## UNKNOWN Arithmetic operation "-" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
631 | require(senderBalance >= amount, "ERC20: transfer amount exceeds balance");
632 | unchecked {
633 |     _balances[sender] = senderBalance - amount;
634 | }
635 | _balances[recipient] += transferAmount;
```



## UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
633 | _balances[sender] = senderBalance - amount;
634 | }
635 | balances[recipient] += transferAmount;
636 | _balances[address(this)] += donateAmount;
```

## UNKNOWN Arithmetic operation "+=" discovered

This plugin produces issues to support false positive discovery within MythX.

SWC-101

Source file

gougou/gg.sol

Locations

```
634 | }
635 | _balances[recipient] += transferAmount;
636 | balances[address(this)] += donateAmount;
637 |
638 | emit Transfer(sender, recipient, transferAmount);
```

LOW

A floating pragma is set.

The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

SWC-103

Source file

gougou/gg.sol

Locations

```
2 | // OpenZeppelin Contracts v4.4.0 (utils/Context.sol)
3 |
4 | pragma solidity ^0.8.0;
5 |
6 | /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

gougou/gg.sol

Locations

```
26 | // OpenZeppelin Contracts v4.4.0 (access/Ownable.sol)
27 |
28 | pragma solidity ^0.8.0
29 |
30 | /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

gougou/gg.sol

Locations

```
100 | // OpenZeppelin Contracts v4.4.0 (token/ERC20/IERC20.sol)
101 |
102 | pragma solidity ^0.8.0
103 |
104 | /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

gougou/gg.sol

Locations

```
183 | // OpenZeppelin Contracts v4.4.0 (token/ERC20/extensions/IERC20Metadata.sol)
184 |
185 | pragma solidity ^0.8.0
186 |
187 | /**
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

gougou/gg.sol

Locations

```
208 |
209 |
210 | pragma solidity ^0.8.0
211 |
212 | contract ERC20 is Context, IERC20, IERC20Metadata {
```

LOW

A floating pragma is set.

SWC-103

The current pragma Solidity directive is `""^0.8.0""`. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source file

gougou/gg.sol

Locations

```
533 | }
534 |
535 | pragma solidity ^0.8.0
536 |
537 | contract GGToken is Ownable, ERC20 {
```

LOW

State variable visibility is not set.

SWC-108

It is best practice to set the visibility of state variables explicitly. The default visibility for `"_balances"` is internal. Other possible visibility settings are public and private.

Source file

gougou/gg.sol

Locations

```
211 |
212 | contract ERC20 is Context, IERC20, IERC20Metadata {
213 |     mapping(address => uint256) _balances;
214 |
215 |     mapping(address => mapping(address => uint256)) private _allowances;
```