Golden Rabbit Inu

Smart Contract Audit Report

SYSFIXED

13 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Golden Rabbit Inu	GRINU	Binance Smart Chain

Addresses

Contract address	0xbfC0907554c71f7C9c93cDC8e0D5819D340F78A9
Contract deployer address	0x687E5Fef69B473915E95B22d1eBa035e14b9Aad9

Project Website

https://goldenrabbitinu.com/

Codebase

https://bscscan.com/address/0xbfC0907554c71f7C9c93cDC8e0D5819D340F78A9#code

SUMMARY

$GRINU aims to build its own swap with a minimal to no fee, NFT's mystery box with rarity, a Defi app and much more! $GRINU is a Community driven meme coin on the Binance Smart chain that aims to bring the golden days of BNB chain back through community engagement and utilities. Come Join us!

Contract Summary

Documentation Quality

Golden Rabbit Inu provides a very good documentation with standard of solidity base code.

- The technical description is provided clearly and structured and also dont have any high risk issue.

Code Quality

The Overall quality of the basecode is standard.

- Standard solidity basecode and rules are already followed by Golden Rabbit Inu with the discovery of several low issues.

Test Coverage

Test coverage of the project is 100% (Through Codebase)

Audit Findings Summary

- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 148, 149, 151, 152, 153, 154, 276, 282, 292, 337, 352, 354, 382, 394, 400, 401, 405, 408, 410, 414, 417, 419, 423, 426, 428, 432, 435, 437, 478, 479, 480, 481, 482, 484, 509, 515, 516, 517, 518, 519, 521, 545, 546, 552, 558, 561, 562, 564, 599, 610, 615, 651, 657, 661, 664, 665, 673, 683, 692, 693, 699, 700, 701, 708, 713, 718, 758, 780, 786, 797, 798, 799 and 354.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 10.
- SWC-110 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 353, 354, 559, 561, 562, 742, 743 and 759.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 325 and 651.

CONCLUSION

We have audited the Golden Rabbit Inu project released on January 2023 to discover issues and identify potential security vulnerabilities in Golden Rabbit Inu Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

The security audit report provides a satisfactory result with some low-risk issues.

The issues found in the code on Golden Rabbit Inu smart contract do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a floating pragma is set, weak sources of randomness and out of bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	PASS
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	ISSUE FOUND
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
SELFDESTRUCT Instruction	SWC-106	The contract should not be self-destructible while it has funds belonging to users.	PASS
Reentrancy	SWC-107	Check effect interaction pattern should be followed if the code performs recursive call.	PASS
Assert Violation	SWC-110	Properly functioning code should never reach a failing assert statement.	ISSUE FOUND
Deprecated Solidity Functions	SWC-111	Deprecated built-in functions should never be used.	PASS
Delegate call to Untrusted Callee	SWC-112	Delegate calls should only be allowed to trusted addresses.	PASS
DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS

Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	PASS
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	ISSUE FOUND
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they have identical functions, a developer should carefully specify inheritance in the correct order. The rule of thumb is to inherit contracts from more /general/ to more /specific/.	PASS

SMART CONTRACT ANALYSIS

Started	Thursday Jan 12 2023 00:28:37 GMT+0000 (Coordinated Universal Time)
Finished	Friday Jan 13 2023 21:21:29 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	GoldenRabbitInu.sol

Detected Issues

ID	Title	Severity	Status
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "++" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "-=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "++" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "++" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 148

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
147
148   uint256 private _tTotal = 100_000_000 * 10**_decimals;
149   uint256 private _rTotal = (MAX - (MAX % _tTotal));
150
151   uint256 public swapTokensAtAmount = 125_000 * 10**9;
152
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 149

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
148   uint256 private _tTotal = 100_000_000 * 10**_decimals;
149   uint256 private _rTotal = (MAX - (MAX % _tTotal));
150
151   uint256 public swapTokensAtAmount = 125_000 * 10**9;
152   uint256 public maxBuyLimit = 1_000_000 * 10**9;
153
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 151

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
150
151   uint256 public swapTokensAtAmount = 125_000 * 10**9;
152   uint256 public maxBuyLimit = 1_000_000 * 10**9;
153   uint256 public maxSellLimit = 1_000_000 * 10**9;
154   uint256 public maxWalletLimit = 1_500_000 * 10**9;
155
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 152

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
151   uint256 public swapTokensAtAmount = 125_000 * 10**9;
152   uint256 public maxBuyLimit = 1_000_000 * 10**9;
153   uint256 public maxSellLimit = 1_000_000 * 10**9;
154   uint256 public maxWalletLimit = 1_500_000 * 10**9;
155
156
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 153

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
152   uint256 public maxBuyLimit = 1_000_000 * 10**9;
153   uint256 public maxSellLimit = 1_000_000 * 10**9;
154   uint256 public maxWalletLimit = 1_500_000 * 10**9;
155
156   uint256 public genesis_block;
157
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 154

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
153   uint256 public maxSellLimit = 1_000_000 * 10**9;
154   uint256 public maxWalletLimit = 1_500_000 * 10**9;
155
156   uint256 public genesis_block;
157   uint256 private deadline = 0;
158
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 276

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
275   require(currentAllowance >= amount, "BEP20: transfer amount exceeds allowance");
276   _approve(sender, _msgSender(), currentAllowance - amount);
277
278   return true;
279   }
280
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 282

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
281   function increaseAllowance(address spender, uint256 addedValue) public returns
(bool) {
282   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
283   return true;
284   }
285
286
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 292

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
291   require(currentAllowance >= subtractedValue, "BEP20: decreased allowance below
zero");
292   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
293
294   return true;
295   }
296
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 337

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
336   uint256 currentRate = _getRate();
337   return rAmount / currentRate;
338   }
339
340   //reflections = reward
341
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 352

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
351   require(_isExcluded[account], "Account is not excluded");
352   for (uint256 i = 0; i < _excluded.length; i++) {
353   if (_excluded[i] == account) {
354   _excluded[i] = _excluded[_excluded.length - 1];
355   _tOwned[account] = 0;
356
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 354

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
353   if (_excluded[i] == account) {
354   _excluded[i] = _excluded[_excluded.length - 1];
355   _tOwned[account] = 0;
356   _isExcluded[account] = false;
357   _excluded.pop();
358
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 382

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
381   ) public onlyOwner {
382   require((_reflections + _marketing + _liquidity + _operations + _misc) <= 10, "Fees
can't exceed 10%");
383   taxes = Taxes(_reflections, _marketing, _liquidity, _operations , _misc);
384   emit FeesChanged();
385   }
386
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 394

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
393   ) public onlyOwner {
394   require((_reflections + _marketing + _liquidity + _operations + _misc) <= 10, "Fees
can't exceed 10%");
395   sellTaxes = Taxes(_reflections, _marketing, _liquidity, _operations, _misc);
396   emit FeesChanged();
397   }
398
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 400

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
399   function _reflectreflections(uint256 rreflections, uint256 treflections) private {
400   _rTotal -= rreflections;
401   totFeesPaid.reflections += treflections;
402   }
403
404
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 401

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
400   _rTotal -= rreflections;
401   totFeesPaid.reflections += treflections;
402   }
403
404   function _takeLiquidity(uint256 rLiquidity, uint256 tLiquidity) private {
405
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 405

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
404   function _takeLiquidity(uint256 rLiquidity, uint256 tLiquidity) private {
405   totFeesPaid.liquidity += tLiquidity;
406
407   if (_isExcluded[address(this)]) {
408   _tOwned[address(this)] += tLiquidity;
409
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 408

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
407   if (_isExcluded[address(this)]) {
408   _tOwned[address(this)] += tLiquidity;
409   }
410   _rOwned[address(this)] += rLiquidity;
411   }
412
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 410

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
409   }
410   _rOwned[address(this)] += rLiquidity;
411   }
412
413   function _takeMarketing(uint256 rMarketing, uint256 tMarketing) private {
414
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 414

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
413   function _takeMarketing(uint256 rMarketing, uint256 tMarketing) private {
414   totFeesPaid.marketing += tMarketing;
415
416   if (_isExcluded[address(this)]) {
417   _tOwned[address(this)] += tMarketing;
418
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 417

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
416   if (_isExcluded[address(this)]) {
417   _tOwned[address(this)] += tMarketing;
418   }
419   _rOwned[address(this)] += rMarketing;
420   }
421
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 419

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
418   }
419   _rOwned[address(this)] += rMarketing;
420   }
421
422   function _takeoperations(uint256 roperations, uint256 toperations) private {
423
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 423

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
422   function _takeoperations(uint256 roperations, uint256 toperations) private {
423   totFeesPaid.operations += toperations;
424
425   if (_isExcluded[address(this)]) {
426   _tOwned[address(this)] += toperations;
427
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 426

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
425   if (_isExcluded[address(this)]) {
426   _tOwned[address(this)] += toperations;
427   }
428   _rOwned[address(this)] += roperations;
429   }
430
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 428

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
427   }
428   _rOwned[address(this)] += roperations;
429   }
430
431   function _takemisc(uint256 rmisc, uint256 tmisc) private {
432
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 432

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
431   function _takemisc(uint256 rmisc, uint256 tmisc) private {
432   totFeesPaid.misc += tmisc;
433
434   if (_isExcluded[address(this)]) {
435   _tOwned[address(this)] += tmisc;
436
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 435

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
434   if (_isExcluded[address(this)]) {
435   _tOwned[address(this)] += tmisc;
436   }
437   _rOwned[address(this)] += rmisc;
438   }
439
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 437

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
436   }
437   _rOwned[address(this)] += rmisc;
438   }
439
440   function _getValues(
441
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 478

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
477
478   s.treflections = (tAmount * temp.reflections) / 100;
479   s.tMarketing = (tAmount * temp.marketing) / 100;
480   s.tLiquidity = (tAmount * temp.liquidity) / 100;
481   s.toperations = (tAmount * temp.operations) / 100;
482
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
478   s.treflections = (tAmount * temp.reflections) / 100;
479   s.tMarketing = (tAmount * temp.marketing) / 100;
480   s.tLiquidity = (tAmount * temp.liquidity) / 100;
481   s.toperations = (tAmount * temp.operations) / 100;
482   s.tmisc = (tAmount * temp.misc) / 100;
483
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 480

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
479   s.tMarketing = (tAmount * temp.marketing) / 100;
480   s.tLiquidity = (tAmount * temp.liquidity) / 100;
481   s.toperations = (tAmount * temp.operations) / 100;
482   s.tmisc = (tAmount * temp.misc) / 100;
483   s.tTransferAmount =
484
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 481

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
480   s.tLiquidity = (tAmount * temp.liquidity) / 100;
481   s.toperations = (tAmount * temp.operations) / 100;
482   s.tmisc = (tAmount * temp.misc) / 100;
483   s.tTransferAmount =
484   tAmount -
485
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 482

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
481   s.toperations = (tAmount * temp.operations) / 100;
482   s.tmisc = (tAmount * temp.misc) / 100;
483   s.tTransferAmount =
484   tAmount -
485   s.treflections -
486
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 484

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
483   s.tTransferAmount =
484   tAmount -
485   s.treflections -
486   s.tMarketing -
487   s.tLiquidity -
488
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 509

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
508   {
509   rAmount = tAmount * currentRate;
510
511   if (!takeFee) {
512   return (rAmount, rAmount, 0, 0, 0);
513
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 515

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
514
515   rreflections = s.treflections * currentRate;
516   rMarketing = s.tMarketing * currentRate;
517   rLiquidity = s.tLiquidity * currentRate;
518   uint256 roperations = s.toperations * currentRate;
519
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 516

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
515   rreflections = s.treflections * currentRate;
516   rMarketing = s.tMarketing * currentRate;
517   rLiquidity = s.tLiquidity * currentRate;
518   uint256 roperations = s.toperations * currentRate;
519   uint256 rmisc = s.tmisc * currentRate;
520
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 517

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
516   rMarketing = s.tMarketing * currentRate;
517   rLiquidity = s.tLiquidity * currentRate;
518   uint256 roperations = s.toperations * currentRate;
519   uint256 rmisc = s.tmisc * currentRate;
520   rTransferAmount =
521
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 518

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
517   rLiquidity = s.tLiquidity * currentRate;
518   uint256 roperations = s.toperations * currentRate;
519   uint256 rmisc = s.tmisc * currentRate;
520   rTransferAmount =
521   rAmount -
522
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 519

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
518   uint256 roperations = s.toperations * currentRate;
519   uint256 rmisc = s.tmisc * currentRate;
520   rTransferAmount =
521   rAmount -
522   rreflections -
523
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 521

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
520   rTransferAmount =
521   rAmount -
522   rreflections -
523   rMarketing -
524   rLiquidity -
525
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 545

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
544
545   roperations = s.toperations * currentRate;
546   rmisc = s.tmisc * currentRate;
547   return (roperations,rmisc);
548   }
549
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 546

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
545   roperations = s.toperations * currentRate;
546   rmisc = s.tmisc * currentRate;
547   return (roperations,rmisc);
548   }
549
550
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 552

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
551   (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
552   return rSupply / tSupply;
553   }
554
555   function _getCurrentSupply() private view returns (uint256, uint256) {
556
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 558

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
557   uint256 tSupply = _tTotal;
558   for (uint256 i = 0; i < _excluded.length; i++) {
559   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply)
560   return (_rTotal, _tTotal);
561   rSupply = rSupply - _rOwned[_excluded[i]];
562
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 561

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
560   return (_rTotal, _tTotal);
561   rSupply = rSupply - _rOwned[_excluded[i]];
562   tSupply = tSupply - _tOwned[_excluded[i]];
563   }
564   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
565
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 562

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
561   rSupply = rSupply - _rOwned[_excluded[i]];
562   tSupply = tSupply - _tOwned[_excluded[i]];
563   }
564   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
565   return (rSupply, tSupply);
566
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 564

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
563   }
564   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
565   return (rSupply, tSupply);
566   }
567
568
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 599

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
598   require(
599   balanceOf(to) + amount <= maxWalletLimit,
600   "You are exceeding maxWalletLimit"
601   );
602   }
603
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 610

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
609   require(
610   balanceOf(to) + amount <= maxWalletLimit,
611   "You are exceeding maxWalletLimit"
612   );
613   }
614
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 615

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
614   if (coolDownEnabled) {
615   uint256 timePassed = block.timestamp - _lastSell[from];
616   require(timePassed >= coolDownTime, "Cooldown enabled");
617   _lastSell[from] = block.timestamp;
618   }
619
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 651

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
650   !_isExcludedFromFee[recipient] &&
651   block.number < genesis_block + deadline;
652
653   valuesFromGetValues memory s = _getValues(tAmount, takeFee, isSell, useLaunchTax);
654
655
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 657

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
656   //from excluded
657   _tOwned[sender] = _tOwned[sender] - tAmount;
658   }
659   if (_isExcluded[recipient]) {
660   //to excluded
661
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 661

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
660   //to excluded
661   _tOwned[recipient] = _tOwned[recipient] + s.tTransferAmount;
662   }
663
664   _rOwned[sender] = _rOwned[sender] - s.rAmount;
665
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 664

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
663
664   _rOwned[sender] = _rOwned[sender] - s.rAmount;
665   _rOwned[recipient] = _rOwned[recipient] + s.rTransferAmount;
666
667   if (s.rreflections > 0 || s.treflections > 0) _reflectreflections(s.rreflections,
s.treflections);
668
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 665

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
664   _rOwned[sender] = _rOwned[sender] - s.rAmount;
665   _rOwned[recipient] = _rOwned[recipient] + s.rTransferAmount;
666
667   if (s.rreflections > 0 || s.treflections > 0) _reflectreflections(s.rreflections,
s.treflections);
668   if (s.rLiquidity > 0 || s.tLiquidity > 0) {
669
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 673

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
672   address(this),
673   s.tLiquidity + s.tMarketing + s.toperations + s.tmisc
674   );
675   }
676   if (s.rMarketing > 0 || s.tMarketing > 0) _takeMarketing(s.rMarketing,
s.tMarketing);
677
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 683

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
682   function swapAndLiquify(uint256 contractBalance, Taxes memory temp) private
lockTheSwap {
683   uint256 denominator = (temp.liquidity +
684   temp.marketing +
685   temp.operations +
686   temp.misc) * 2;
687
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 692

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
691
692   uint256 tokensToAddLiquidityWith = (contractBalance * temp.liquidity) /
denominator;
693   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
694
695   uint256 initialBalance = address(this).balance;
696
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 693

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
692   uint256 tokensToAddLiquidityWith = (contractBalance * temp.liquidity) /
denominator;
693   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
694
695   uint256 initialBalance = address(this).balance;
696
697
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 699

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
698
699   uint256 deltaBalance = address(this).balance - initialBalance;
700   uint256 unitBalance = deltaBalance / (denominator - temp.liquidity);
701   uint256 bnbToAddLiquidityWith = unitBalance * temp.liquidity;
702
703
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 700

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
699   uint256 deltaBalance = address(this).balance - initialBalance;
700   uint256 unitBalance = deltaBalance / (denominator - temp.liquidity);
701   uint256 bnbToAddLiquidityWith = unitBalance * temp.liquidity;
702
703   if (bnbToAddLiquidityWith > 0) {
704
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 701

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
700   uint256 unitBalance = deltaBalance / (denominator - temp.liquidity);
701   uint256 bnbToAddLiquidityWith = unitBalance * temp.liquidity;
702
703   if (bnbToAddLiquidityWith > 0) {
704   // Add liquidity to pancake
705
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 708

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
707
708   uint256 marketingAmt = unitBalance * 2 * temp.marketing;
709   if (marketingAmt > 0) {
710   payable(marketingWallet).sendValue(marketingAmt);
711   }
712
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 708

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
707
708   uint256 marketingAmt = unitBalance * 2 * temp.marketing;
709   if (marketingAmt > 0) {
710   payable(marketingWallet).sendValue(marketingAmt);
711   }
712
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 713

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
712
713   uint256 operationsAmt = unitBalance * 2 * temp.operations;
714   if (operationsAmt > 0) {
715   payable(operationsWallet).sendValue(operationsAmt);
716   }
717
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 718

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
717
718   uint256 miscAmt = unitBalance * 2 * temp.misc;
719   if (miscAmt > 0) {
720   payable(miscWallet).sendValue(miscAmt);
721   }
722
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 758

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
757   function bulkExcludeFee(address[] memory accounts, bool state) external onlyOwner {
758   for (uint256 i = 0; i < accounts.length; i++) {
759   _isExcludedFromFee[accounts[i]] = state;
760   }
761   }
762
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 780

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
779   require(time <= 60, "cooldown timer cannot exceed 1 minutes");
780   coolDownTime = time * 1 seconds;
781   coolDownEnabled = state;
782   }
783
784
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 786

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
785   require(amount <= 1_000_000, "Cannot set swap threshold amount higher than 1% of
tokens");
786   swapTokensAtAmount = amount * 10**_decimals;
787   }
788
789   function updateSwapEnabled(bool _enabled) external onlyOwner {
790
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 797

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
796   require(maxWallet >= 1_000_000, "Cannot set max wallet amount lower than 1% of
total supply");
797   maxBuyLimit = maxBuy * 10**decimals();
798   maxSellLimit = maxSell * 10**decimals();
799   maxWalletLimit = maxWallet * 10**decimals();
800   }
801
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 798

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
797   maxBuyLimit = maxBuy * 10**decimals();
798   maxSellLimit = maxSell * 10**decimals();
799   maxWalletLimit = maxWallet * 10**decimals();
800   }
801
802
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 799

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
798   maxSellLimit = maxSell * 10**decimals();
799   maxWalletLimit = maxWallet * 10**decimals();
800   }
801
802   function updateRouterAndPair(address newRouter, address newPair) external onlyOwner
{
803
```

SWC-101 | COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED

LINE 354

low SEVERITY

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

Source File

- GoldenRabbitInu.sol

Locations

```
353   if (_excluded[i] == account) {
354   _excluded[i] = _excluded[_excluded.length - 1];
355   _tOwned[account] = 0;
356   _isExcluded[account] = false;
357   _excluded.pop();
358
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 10

low SEVERITY

The current pragma Solidity directive is ""^0.8.17"". 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

- GoldenRabbitInu.sol

Locations

```
9   // SPDX-License-Identifier: UNLICENSE
10   pragma solidity ^0.8.17;
11
12   interface IBEP20 {
13   function totalSupply() external view returns (uint256);
14
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 353

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
352   for (uint256 i = 0; i < _excluded.length; i++) {
353   if (_excluded[i] == account) {
354   _excluded[i] = _excluded[_excluded.length - 1];
355   _tOwned[account] = 0;
356   _isExcluded[account] = false;
357
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 354

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
353   if (_excluded[i] == account) {
354   _excluded[i] = _excluded[_excluded.length - 1];
355   _tOwned[account] = 0;
356   _isExcluded[account] = false;
357   _excluded.pop();
358
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 559

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
558   for (uint256 i = 0; i < _excluded.length; i++) {
559   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply)
560   return (_rTotal, _tTotal);
561   rSupply = rSupply - _rOwned[_excluded[i]];
562   tSupply = tSupply - _tOwned[_excluded[i]];
563
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 561

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
560   return (_rTotal, _tTotal);
561   rSupply = rSupply - _rOwned[_excluded[i]];
562   tSupply = tSupply - _tOwned[_excluded[i]];
563   }
564   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
565
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 562

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
561   rSupply = rSupply - _rOwned[_excluded[i]];
562   tSupply = tSupply - _tOwned[_excluded[i]];
563   }
564   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
565   return (rSupply, tSupply);
566
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 742

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
741   address[] memory path = new address[](2);
742   path[0] = address(this);
743   path[1] = router.WETH();
744
745   _approve(address(this), address(router), tokenAmount);
746
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 743

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
742   path[0] = address(this);
743   path[1] = router.WETH();
744
745   _approve(address(this), address(router), tokenAmount);
746
747
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 759

low SEVERITY

The index access expression can cause an exception in case of use of invalid array index value.

Source File

- GoldenRabbitInu.sol

Locations

```
758   for (uint256 i = 0; i < accounts.length; i++) {
759   _isExcludedFromFee[accounts[i]] = state;
760   }
761   }
762
763
```

SWC-120 | POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.

LINE 325

low SEVERITY

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source File

- GoldenRabbitInu.sol

Locations

```
324   swapEnabled = true;
325   genesis_block = block.number;
326   }
327
328   function updatedeadline(uint256 _deadline) external onlyOwner {
329
```

SWC-120 | POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.

LINE 651

low SEVERITY

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

Source File

- GoldenRabbitInu.sol

Locations

```
650   !_isExcludedFromFee[recipient] &&
651   block.number < genesis_block + deadline;
652
653   valuesFromGetValues memory s = _getValues(tAmount, takeFee, isSell, useLaunchTax);
654
655
```

DISCLAIMER

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to, or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without Sysfixed's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Sysfixed to perform a security assessment. This report does 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, business, business model, or legal compliance.

This is a limited report on our findings based on our analysis, in accordance with good industry practice as of the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing 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.

This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and Sysfixed and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers, and other representatives) (Sysfixed) owe no duty of care.

ABOUT US

Sysfixed is a blockchain security certification organization established in 2021 with the objective to provide smart contract security services and verify their correctness in blockchain-based protocols. Sysfixed automatically scans for security vulnerabilities in Ethereum and other EVM-based blockchain smart contracts. Sysfixed a comprehensive range of analysis techniques—including static analysis, dynamic analysis, and symbolic execution—can accurately detect security vulnerabilities to provide an in-depth analysis report. With a vibrant ecosystem of world-class integration partners that amplify developer productivity, Sysfixed can be utilized in all phases of your project's lifecycle. Our team of security experts is dedicated to the research and improvement of our tools and techniques used to fortify your code.