



Dogcoin

Smart Contract Audit Report

TABLE OF CONTENTS

[Audited Details](#)

- Audited Project
- Blockchain
- Addresses
- Project Website
- Codebase

[Summary](#)

- Contract Summary
- Audit Findings Summary
- Vulnerabilities Summary

[Conclusion](#)

[Audit Results](#)

[Smart Contract Analysis](#)

- Detected Vulnerabilities

[Disclaimer](#)

[About Us](#)

AUDITED DETAILS

Audited Project

| Project name | Token ticker | Blockchain |
|--------------|--------------|---------------------|
| Dogcoin | DOGS | Binance Smart Chain |

Addresses

| | |
|---------------------------|--|
| Contract address | 0xbccd27062ae1a2bea5731c904b96edfb163aba21 |
| Contract deployer address | 0x9F2f0f30D7446668D9c1ae94CC29b8F70817a8AE |

Project Website

| |
|---|
| https://dogcoin.network/ |
|---|

Codebase

| |
|---|
| https://bscscan.com/address/0xbccd27062ae1a2bea5731c904b96edfb163aba21#code |
|---|

SUMMARY

The Dogcoin Platform is an EVM compatible blockchain that bridges the gap between technical, traffic and ecological resources. It offers enhanced efficiency and costeffectiveness when constructing decentralised applications, as well as comprehensive support in promotion, traffic and resource management. This unified infrastructure platform is positioned to revolutionise the DApp industry. Developers around the world will be able to access a variety of innovative facilities and services delivered by Dogcoin.

Contract Summary

Documentation Quality

Dogcoin 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 Dogcoin 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 120, 120, 121, 121, 122, 128, 128, 129, 129, 132, 132, 244, 250, 257, 268, 279, 280, 288, 290, 293, 297, 297, 299, 299, 304, 308, 335, 336, 340, 392, 394, 395, 411, 411, 411, 426, 426, 428, 429, 435, 435, 437, 438, 445, 445, 447, 448, 456, 456, 458, 459, 459, 460, 478, 482, 526, 526, 530, 530, 543, 553, 565, 566, 566, 583, 584, 585, 585, 587, 587, 593, 594, 595, 595, 597, 597, 603, 604, 604 and 290.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 6.
- SWC-110 SWC-123 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 289, 290, 290, 504, 505, 544, 554, 604 and 604.

CONCLUSION

We have audited the Dogcoin project released on August 2022 to discover issues and identify potential security vulnerabilities in Dogcoin 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 satisfactory results with low-risk issues.

The Dogcoin smart contract code issues 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 a floating pragma set and out-of-bounds array access which the index access expression can cause an exception in case of an invalid array index value. The current pragma Solidity directive is `^0.8.7`. 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.

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 |
| Unprotected Ether Withdrawal | SWC-105 | Due to missing or insufficient access controls, malicious parties can withdraw from the contract. | 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 |
| Uninitialized Storage Pointer | SWC-109 | Uninitialized local storage variables can point to unexpected storage locations in the contract. | PASS |
| Assert Violation | SWC-110 SWC-123 | 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 | Delegatecalls 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 |
| Incorrect Constructor Name | SWC-118 | Constructors are special functions that are called only once during the contract creation. | 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. | PASS |
| Write to Arbitrary Storage Location | SWC-124 | The contract is responsible for ensuring that only authorized user or contract accounts may write to sensitive storage locations. | PASS |
| 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 |
| Insufficient Gas Griefing | SWC-126 | Insufficient gas grieving attacks can be performed on contracts which accept data and use it in a sub-call on another contract. | PASS |
| Arbitrary Jump Function | SWC-127 | As Solidity doesnt support pointer arithmetics, it is impossible to change such variable to an arbitrary value. | PASS |

| | | | |
|----------------------------|--------------------|--|------|
| Typographical Error | SWC-129 | A typographical error can occur for example when the intent of a defined operation is to sum a number to a variable. | PASS |
| Override control character | SWC-130 | Malicious actors can use the Right-To-Left-Override unicode character to force RTL text rendering and confuse users as to the real intent of a contract. | PASS |
| Unused variables | SWC-131 SWC-135 | Unused variables are allowed in Solidity and they do not pose a direct security issue. | PASS |
| Unexpected Ether balance | SWC-132 | Contracts can behave erroneously when they strictly assume a specific Ether balance. | PASS |
| Hash Collisions Variable | SWC-133 | Using abi.encodePacked() with multiple variable length arguments can, in certain situations, lead to a hash collision. | PASS |
| Hardcoded gas amount | SWC-134 | The transfer() and send() functions forward a fixed amount of 2300 gas. | PASS |
| Unencrypted Private Data | SWC-136 | It is a common misconception that private type variables cannot be read. | PASS |

SMART CONTRACT ANALYSIS

| | |
|------------------|---|
| Started | Sunday Aug 14 2022 09:11:43 GMT+0000 (Coordinated Universal Time) |
| Finished | Monday Aug 15 2022 22:39:50 GMT+0000 (Coordinated Universal Time) |
| Mode | Standard |
| Main Source File | Dogcoin.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

LINE 120

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
119
120 uint256 private constant T_TOTAL = 1e15 * 10**DECIMALS;
121 uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
122 uint256 private _reflectionRate = _rTotal / T_TOTAL;
123 uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
124
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 120

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
119
120 uint256 private constant T_TOTAL = 1e15 * 10**DECIMALS;
121 uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
122 uint256 private _reflectionRate = _rTotal / T_TOTAL;
123 uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
124
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 121

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
120 uint256 private constant T_TOTAL = 1e15 * 10**DECIMALS;  
121 uint256 private _rTotal = (MAX - (MAX % T_TOTAL));  
122 uint256 private _reflectionRate = _rTotal / T_TOTAL;  
123 uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;  
124 uint256 private _tExcludedFromRewardTotal;  
125
```


SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 121

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
120 uint256 private constant T_TOTAL = 1e15 * 10**DECIMALS;  
121 uint256 private _rTotal = (MAX - (MAX % T_TOTAL));  
122 uint256 private _reflectionRate = _rTotal / T_TOTAL;  
123 uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;  
124 uint256 private _tExcludedFromRewardTotal;  
125
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 122

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
121 uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
122 uint256 private _reflectionRate = _rTotal / T_TOTAL;
123 uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
124 uint256 private _tExcludedFromRewardTotal;
125 uint256 private _rExcludedFromRewardTotal;
126
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 128

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
127
128  uint256 public antiWhaleAmt = 50_000_000_000_000 * 10**DECIMALS;
129  uint256 public swapTokensAtAmount = 20_000_000_000_000 * 10**DECIMALS;
130
131  // Anti Dump //
132
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 128

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
127
128  uint256 public antiWhaleAmt = 50_000_000_000_000 * 10**DECIMALS;
129  uint256 public swapTokensAtAmount = 20_000_000_000_000 * 10**DECIMALS;
130
131  // Anti Dump //
132
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 129

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
128 uint256 public antiWhaleAmt = 50_000_000_000_000 * 10**DECIMALS;
129 uint256 public swapTokensAtAmount = 20_000_000_000_000 * 10**DECIMALS;
130
131 // Anti Dump //
132 uint256 public maxSellAmountPerCycle = 50_000_000_000_000 * 10**DECIMALS;
133
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 129

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
128 uint256 public antiWhaleAmt = 50_000_000_000_000 * 10**DECIMALS;  
129 uint256 public swapTokensAtAmount = 20_000_000_000_000 * 10**DECIMALS;  
130  
131 // Anti Dump //  
132 uint256 public maxSellAmountPerCycle = 50_000_000_000_000 * 10**DECIMALS;  
133
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
131 // Anti Dump //
132 uint256 public maxSellAmountPerCycle = 50_000_000_000_000 * 10**DECIMALS;
133 uint256 public antiDumpCycle = 1 hours;
134
135 // only allow Whitelist PancakeSwap Trading //
136
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
131 // Anti Dump //  
132 uint256 public maxSellAmountPerCycle = 50_000_000_000_000 * 10**DECIMALS;  
133 uint256 public antiDumpCycle = 1 hours;  
134  
135 // only allow Whitelist PancakeSwap Trading //  
136
```


SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 244

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
243     require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
244     _approve(sender, _msgSender(), currentAllowance - amount);
245
246     return true;
247 }
248
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 250

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
249     function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
250     _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
251     return true;
252 }
253
254
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 257

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
256     require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
257     _approve(_msgSender(), spender, currentAllowance - subtractedValue);
258
259     return true;
260 }
261
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 268

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
267     require(rAmount <= _rTotal, "Amount must be less than total reflections");
268     return rAmount / _reflectionRate;
269 }
270
271 //@dev kept original RFI naming -> "reward" as in reflection
272
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 279

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
278  _rOwned[account] = 0;  
279  _tExcludedFromRewardTotal += tBalance;  
280  _rExcludedFromRewardTotal += rBalance;  
281  }  
282  _isExcludedFromReward[account] = true;  
283
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 280

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
279     _tExcludedFromRewardTotal += tBalance;  
280     _rExcludedFromRewardTotal += rBalance;  
281 }  
282 _isExcludedFromReward[account] = true;  
283 _excludedFromReward.push(account);  
284
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 288

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
287     require(!_isExcludedFromReward[account], "Account is not excluded");
288     for (uint256 i = 0; i < _excludedFromReward.length; i++) {
289         if (_excludedFromReward[i] == account) {
290             _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
291             uint256 tBalance = _tOwned[account];
292         }
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 290

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
289     if (_excludedFromReward[i] == account) {
290         _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
291         uint256 tBalance = _tOwned[account];
292         if (tBalance > 0) {
293             uint256 rBalance = tBalance * _reflectionRate;
294         }
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 293

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
292   if (tBalance > 0) {  
293     uint256 rBalance = tBalance * _reflectionRate;  
294     if (tBalance == _tExcludedFromRewardTotal) {  
295       // try fix rTotal  
296       if (rBalance > _rExcludedFromRewardTotal) {  
297
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 297

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
296   if (rBalance > _rExcludedFromRewardTotal) {  
297     _rTotal += rBalance - _rExcludedFromRewardTotal;  
298   } else if (rBalance < _rExcludedFromRewardTotal) {  
299     _rTotal -= _rExcludedFromRewardTotal - rBalance;  
300   }  
301
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 297

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
296   if (rBalance > _rExcludedFromRewardTotal) {  
297     _rTotal += rBalance - _rExcludedFromRewardTotal;  
298   } else if (rBalance < _rExcludedFromRewardTotal) {  
299     _rTotal -= _rExcludedFromRewardTotal - rBalance;  
300   }  
301
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 299

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
298     } else if (rBalance < _rExcludedFromRewardTotal) {  
299         _rTotal -= _rExcludedFromRewardTotal - rBalance;  
300     }  
301     _rExcludedFromRewardTotal = 0;  
302  
303
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 299

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
298     } else if (rBalance < _rExcludedFromRewardTotal) {  
299         _rTotal -= _rExcludedFromRewardTotal - rBalance;  
300     }  
301     _rExcludedFromRewardTotal = 0;  
302  
303
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 304

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
303     } else {  
304         _rExcludedFromRewardTotal -= rBalance;  
305     }  
306     _tOwned[account] = 0;  
307     _rOwned[account] = rBalance;  
308
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 308

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
307  _rOwned[account] = rBalance;  
308  _tExcludedFromRewardTotal -= tBalance;  
309  }  
310  _isExcludedFromReward[account] = false;  
311  _excludedFromReward.pop();  
312
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 335

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
334
335  uint256 tSupply = T_TOTAL - _tExcludedFromRewardTotal;
336  uint256 rSupply = _rTotal - _rExcludedFromRewardTotal;
337  if (tSupply == 0) {
338    return;
339
```


SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 336

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
335  uint256 tSupply = T_TOTAL - _tExcludedFromRewardTotal;  
336  uint256 rSupply = _rTotal - _rExcludedFromRewardTotal;  
337  if (tSupply == 0) {  
338    return;  
339  }  
340
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 340

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
339     }
340     uint256 newRate = rSupply / tSupply;
341     if (newRate < MIN_REFLECTION_RATE) {
342         _reflectionRate = MIN_REFLECTION_RATE;
343         isRewardEnded = true;
344     }
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 392

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
391     if(transferType == ETransferType.Sell && !traderExcludedFromFee) {  
392         bool newCycle = block.timestamp - userLastSell[from].lastSellTime >= antiDumpCycle;  
393         if(!newCycle){  
394             require(userLastSell[from].amountSoldInCycle + amount <= maxSellAmountPerCycle,  
"You are exceeding maxSellAmountPerCycle");  
395             userLastSell[from].amountSoldInCycle += amount;  
396         }
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 394

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
393     if(!newCycle){
394         require(userLastSell[from].amountSoldInCycle + amount <= maxSellAmountPerCycle,
"You are exceeding maxSellAmountPerCycle");
395         userLastSell[from].amountSoldInCycle += amount;
396     }
397     else{
398
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 395

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
394     require(userLastSell[from].amountSoldInCycle + amount <= maxSellAmountPerCycle,  
    "You are exceeding maxSellAmountPerCycle");  
395     userLastSell[from].amountSoldInCycle += amount;  
396 }  
397 else{  
398     require(amount <= maxSellAmountPerCycle, "You are exceeding  
maxSellAmountPerCycle");  
399
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 411

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
410     traderExcludedFromFee ||  
411     usedTaxes.rfi + usedTaxes.marketing + usedTaxes.liquidity + usedTaxes.burn == 0) {  
412     taxFreeTransfer(from, to, amount);  
413     } else {  
414     _tokenTransfer(from, to, amount, usedTaxes);  
415
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 411

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
410     traderExcludedFromFee ||  
411     usedTaxes.rfi + usedTaxes.marketing + usedTaxes.liquidity + usedTaxes.burn == 0) {  
412     taxFreeTransfer(from, to, amount);  
413     } else {  
414     _tokenTransfer(from, to, amount, usedTaxes);  
415
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 411

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
410     traderExcludedFromFee ||  
411     usedTaxes.rfi + usedTaxes.marketing + usedTaxes.liquidity + usedTaxes.burn == 0) {  
412     taxFreeTransfer(from, to, amount);  
413     } else {  
414     _tokenTransfer(from, to, amount, usedTaxes);  
415
```


SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 426

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
425     if(usedTaxes.liquidity != 0) {  
426         uint256 tLiquidity = tAmount * usedTaxes.liquidity / 100;  
427         if (tLiquidity != 0) {  
428             tTransferAmount -= tLiquidity;  
429             totFeesPaid.liquidity += tLiquidity;  
430         }
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 426

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
425     if(usedTaxes.liquidity != 0) {  
426         uint256 tLiquidity = tAmount * usedTaxes.liquidity / 100;  
427         if (tLiquidity != 0) {  
428             tTransferAmount -= tLiquidity;  
429             totFeesPaid.liquidity += tLiquidity;  
430         }
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 428

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
427     if (tLiquidity != 0) {  
428         tTransferAmount -= tLiquidity;  
429         totFeesPaid.liquidity += tLiquidity;  
430         _addBalance(address(this), tLiquidity, rate);  
431         emit Transfer(sender, address(this), tLiquidity);  
432     }
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 429

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
428     tTransferAmount -= tLiquidity;  
429     totFeesPaid.liquidity += tLiquidity;  
430     _addBalance(address(this), tLiquidity, rate);  
431     emit Transfer(sender, address(this), tLiquidity);  
432 }  
433
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 435

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
434   if (usedTaxes.marketing != 0 && marketingAddress != DEAD_ADDRESS) {  
435       uint256 tMarketing = tAmount * usedTaxes.marketing / 100;  
436       if (tMarketing != 0) {  
437           tTransferAmount -= tMarketing;  
438           totFeesPaid.marketing += tMarketing;  
439       }
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 435

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
434   if (usedTaxes.marketing != 0 && marketingAddress != DEAD_ADDRESS) {  
435       uint256 tMarketing = tAmount * usedTaxes.marketing / 100;  
436       if (tMarketing != 0) {  
437           tTransferAmount -= tMarketing;  
438           totFeesPaid.marketing += tMarketing;  
439       }
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 437

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
436   if (tMarketing != 0) {  
437       tTransferAmount -= tMarketing;  
438       totFeesPaid.marketing += tMarketing;  
439       _addBalance(marketingAddress, tMarketing, rate);  
440       emit Transfer(sender, marketingAddress, tMarketing);  
441   }
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
437     tTransferAmount -= tMarketing;  
438     totFeesPaid.marketing += tMarketing;  
439     _addBalance(marketingAddress, tMarketing, rate);  
440     emit Transfer(sender, marketingAddress, tMarketing);  
441 }  
442
```


SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 445

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
444     if (usedTaxes.burn != 0) {  
445         uint256 tBurn = tAmount * usedTaxes.burn / 100;  
446         if (tBurn != 0) {  
447             tTransferAmount -= tBurn;  
448             totFeesPaid.burn += tBurn;  
449         }
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 445

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
444   if (usedTaxes.burn != 0) {  
445       uint256 tBurn = tAmount * usedTaxes.burn / 100;  
446       if (tBurn != 0) {  
447           tTransferAmount -= tBurn;  
448           totFeesPaid.burn += tBurn;  
449       }
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 447

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
446     if (tBurn != 0) {  
447         tTransferAmount -= tBurn;  
448         totFeesPaid.burn += tBurn;  
449         _addBalance(DEAD_ADDRESS, tBurn, rate);  
450         emit Transfer(sender, DEAD_ADDRESS, tBurn);  
451     }
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 448

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
447     tTransferAmount -= tBurn;
448     totFeesPaid.burn += tBurn;
449     _addBalance(DEAD_ADDRESS, tBurn, rate);
450     emit Transfer(sender, DEAD_ADDRESS, tBurn);
451 }
452
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 456

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
455     if (usedTaxes.rfi != 0 && !isRewardEnded) {  
456         uint256 tRfi = tAmount * usedTaxes.rfi / 100;  
457         if (tRfi != 0) {  
458             tTransferAmount -= tRfi;  
459             _rTotal -= tRfi * _reflectionRate;  
460         }
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 456

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
455     if (usedTaxes.rfi != 0 && !isRewardEnded) {  
456         uint256 tRfi = tAmount * usedTaxes.rfi / 100;  
457         if (tRfi != 0) {  
458             tTransferAmount -= tRfi;  
459             _rTotal -= tRfi * _reflectionRate;  
460         }
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 458

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
457     if (tRfi != 0) {  
458         tTransferAmount -= tRfi;  
459         _rTotal -= tRfi * _reflectionRate;  
460         totFeesPaid.rfi += tRfi;  
461         needRecalcReflectionRate = true;  
462     }
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 459

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
458     tTransferAmount -= tRfi;  
459     _rTotal -= tRfi * _reflectionRate;  
460     totFeesPaid.rfi += tRfi;  
461     needRecalcReflectionRate = true;  
462 }  
463
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 459

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
458     tTransferAmount -= tRfi;  
459     _rTotal -= tRfi * _reflectionRate;  
460     totFeesPaid.rfi += tRfi;  
461     needRecalcReflectionRate = true;  
462 }  
463
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 460

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
459  _rTotal -= tRfi * _reflectionRate;  
460  totFeesPaid.rfi += tRfi;  
461  needRecalcReflectionRate = true;  
462  }  
463  }  
464
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 478

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
477 //calculate how many tokens we need to exchange
478 uint256 tokensToSwap = contractTokenBalance / 2;
479 uint256 otherHalfOfTokens = tokensToSwap;
480 uint256 initialBalance = address(this).balance;
481 swapTokensForBNB(tokensToSwap, address(this));
482
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 482

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
481  swapTokensForBNB(tokensToSwap, address(this));  
482  uint256 newBalance = address(this).balance - (initialBalance);  
483  addLiquidity(otherHalfOfTokens, newBalance);  
484  }  
485  
486
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 526

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
525     function updateAntiWhaleAmt(uint256 amount) external onlyOwner{
526         antiWhaleAmt = amount * 10**DECIMALS;
527     }
528
529     function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
530
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 526

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
525     function updateAntiWhaleAmt(uint256 amount) external onlyOwner{
526         antiWhaleAmt = amount * 10**DECIMALS;
527     }
528
529     function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
530
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 530

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
529     function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
530         swapTokensAtAmount = amount * 10**DECIMALS;
531     }
532
533     function updateSwapEnabled(bool _enabled) external onlyOwner{
534
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 530

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
529     function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
530         swapTokensAtAmount = amount * 10**DECIMALS;
531     }
532
533     function updateSwapEnabled(bool _enabled) external onlyOwner{
534
```


SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 543

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
542 function bulkAntiBot(address[] memory accounts, bool state) external onlyOwner{
543     for(uint256 i = 0; i < accounts.length; i++){
544         _isBot[accounts[i]] = state;
545     }
546 }
547
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 553

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
552     function bulkPancakeSwapWhitelist(address[] memory accounts, bool state) external  
onlyOwner{  
553     for(uint256 i = 0; i < accounts.length; i++){  
554     _isPancakeSwapWhitelisted[accounts[i]] = state;  
555     }  
556     }  
557
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 565

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
564     require(_maxSellAmountPerCycle >= 1_000_000_000, "Amount must be >= 1B");
565     antiDumpCycle = timeInMinutes * 1 minutes;
566     maxSellAmountPerCycle = _maxSellAmountPerCycle * 10**DECIMALS;
567 }
568
569
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 566

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
565     antiDumpCycle = timeInMinutes * 1 minutes;  
566     maxSellAmountPerCycle = _maxSellAmountPerCycle * 10**DECIMALS;  
567 }  
568  
569 function isBot(address account) public view returns(bool){  
570
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 566

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
565     antiDumpCycle = timeInMinutes * 1 minutes;  
566     maxSellAmountPerCycle = _maxSellAmountPerCycle * 10**DECIMALS;  
567 }  
568  
569 function isBot(address account) public view returns(bool){  
570
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 583

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
582     if (!_isExcludedFromReward[account]) {  
583         _tOwned[account] += tAmount;  
584         _tExcludedFromRewardTotal += tAmount;  
585         _rExcludedFromRewardTotal += tAmount * rate;  
586     } else {  
587
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 584

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
583  _tOwned[account] += tAmount;  
584  _tExcludedFromRewardTotal += tAmount;  
585  _rExcludedFromRewardTotal += tAmount * rate;  
586  } else {  
587  _rOwned[account] += tAmount * rate;  
588
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 585

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
584  _tExcludedFromRewardTotal += tAmount;  
585  _rExcludedFromRewardTotal += tAmount * rate;  
586  } else {  
587  _rOwned[account] += tAmount * rate;  
588  }  
589
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 585

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
584     _tExcludedFromRewardTotal += tAmount;  
585     _rExcludedFromRewardTotal += tAmount * rate;  
586 } else {  
587     _rOwned[account] += tAmount * rate;  
588 }  
589
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 587

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
586     } else {  
587         _rOwned[account] += tAmount * rate;  
588     }  
589 }  
590  
591
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 587

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
586     } else {  
587         _rOwned[account] += tAmount * rate;  
588     }  
589 }  
590  
591
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 593

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
592     if (!_isExcludedFromReward[account]) {  
593         _tOwned[account] -= tAmount;  
594         _tExcludedFromRewardTotal -= tAmount;  
595         _rExcludedFromRewardTotal -= tAmount * rate;  
596     } else {  
597
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 594

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
593  _tOwned[account] -= tAmount;  
594  _tExcludedFromRewardTotal -= tAmount;  
595  _rExcludedFromRewardTotal -= tAmount * rate;  
596  } else {  
597  _rOwned[account] -= tAmount * rate;  
598
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 595

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
594  _tExcludedFromRewardTotal -= tAmount;  
595  _rExcludedFromRewardTotal -= tAmount * rate;  
596  } else {  
597  _rOwned[account] -= tAmount * rate;  
598  }  
599
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 595

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
594  _tExcludedFromRewardTotal -= tAmount;  
595  _rExcludedFromRewardTotal -= tAmount * rate;  
596  } else {  
597  _rOwned[account] -= tAmount * rate;  
598  }  
599
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 597

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
596     } else {  
597         _rOwned[account] -= tAmount * rate;  
598     }  
599 }  
600  
601
```


SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 597

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
596     } else {  
597         _rOwned[account] -= tAmount * rate;  
598     }  
599 }  
600  
601
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 603

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
602   require(accounts.length == amounts.length, "Arrays must have the same size");
603   for(uint256 i= 0; i < accounts.length; i++){
604       taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);
605   }
606   }
607
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 604

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
603     for(uint256 i= 0; i < accounts.length; i++){  
604         taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);  
605     }  
606 }  
607  
608
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 604

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
603     for(uint256 i= 0; i < accounts.length; i++){  
604         taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);  
605     }  
606 }  
607  
608
```

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

LINE 290

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
289   if (_excludedFromReward[i] == account) {  
290     _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];  
291     uint256 tBalance = _tOwned[account];  
292     if (tBalance > 0) {  
293       uint256 rBalance = tBalance * _reflectionRate;  
294     }
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

low SEVERITY

The current pragma Solidity directive is `""^0.8.7"`. 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

- Dogcoin.sol

Locations

```
5 // SPDX-License-Identifier: NOLICENSE
6 pragma solidity ^0.8.7;
7
8 interface IERC20 {
9     function totalSupply() external view returns (uint256);
10
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 289

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
288   for (uint256 i = 0; i < _excludedFromReward.length; i++) {  
289     if (_excludedFromReward[i] == account) {  
290       _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];  
291       uint256 tBalance = _tOwned[account];  
292       if (tBalance > 0) {  
293
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 290

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
289     if (_excludedFromReward[i] == account) {  
290         _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];  
291         uint256 tBalance = _tOwned[account];  
292         if (tBalance > 0) {  
293             uint256 rBalance = tBalance * _reflectionRate;  
294         }
```


SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 290

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
289   if (_excludedFromReward[i] == account) {  
290       _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];  
291       uint256 tBalance = _tOwned[account];  
292       if (tBalance > 0) {  
293           uint256 rBalance = tBalance * _reflectionRate;  
294
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 504

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
503     address[] memory path = new address[](2);  
504     path[0] = address(this);  
505     path[1] = router.WETH();  
506  
507     _approve(address(this), address(router), tokenAmount);  
508
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 505

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
504   path[0] = address(this);  
505   path[1] = router.WETH();  
506  
507   _approve(address(this), address(router), tokenAmount);  
508  
509
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 544

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
543   for(uint256 i = 0; i < accounts.length; i++){  
544     _isBot[accounts[i]] = state;  
545   }  
546 }  
547  
548
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 554

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
553     for(uint256 i = 0; i < accounts.length; i++){  
554         _isPancakeSwapWhitelisted[accounts[i]] = state;  
555     }  
556 }  
557  
558
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 604

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
603     for(uint256 i= 0; i < accounts.length; i++){  
604         taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);  
605     }  
606 }  
607  
608
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 604

low SEVERITY

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

Source File

- Dogcoin.sol

Locations

```
603     for(uint256 i= 0; i < accounts.length; i++){  
604         taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);  
605     }  
606 }  
607  
608
```

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.