Catcoin

Smart Contract Audit Report

SYSFIXED

31 Jan 2023

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
Catcoin	CATCOIN	Binance Smart Chain

Addresses

Contract address	0x06df77854793849f770b6af0af4b22511df53a11
Contract deployer address	0x521A6B49569112c8b793Ee545a4df4daE0fda54A

Project Website

https://catcoin.com/

Codebase

https://bscscan.com/address/0x06df77854793849f770b6af0af4b22511df53a11#code

SUMMARY

Catcoin is a digital asset that uses cryptography to secure transactions. It is developed on Binance. Catcoin is the first decentralized web3 project to bring a share-to-earn concept among crypto users where users will earn catcoin tokens for sharing news. Catcoin is launched with the aim of fighting centralized news media that control everything and spread biased news without revealing the complete truth about any news.

Contract Summary

Documentation Quality

Catcoin 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 Catcoin 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 124, 124, 125, 125, 126, 132, 132, 133, 133, 136, 136, 255, 262, 269, 280, 291, 292, 300, 302, 305, 309, 309, 311, 311, 316, 320, 347, 348, 352, 404, 406, 407, 423, 423, 423, 438, 438, 440, 441, 447, 447, 449, 450, 457, 457, 459, 460, 468, 468, 470, 471, 471, 472, 490, 494, 538, 538, 542, 542, 555, 565, 577, 578, 578, 595, 596, 597, 597, 599, 599, 605, 606, 607, 607, 609, 609, 615, 616, 616 and 302.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 10.
- 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 301, 302, 302, 516, 517, 556, 566, 616 and 616.

CONCLUSION

We have audited the Catcoin project released on January 2023 to discover issues and identify potential security vulnerabilities in Catcoin 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 issues found in the Catcoin smart contract code 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 a floating pragma is set 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. 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 griefing 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	Monday Jan 30 2023 16:58:42 GMT+0000 (Coordinated Universal Time)
Finished	Tuesday Jan 31 2023 06:06:43 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	Catcoin.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	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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 124

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
123
124   uint256 private constant T_TOTAL = 1e17 * 10**DECIMALS;
125   uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
126   uint256 private _reflectionRate = _rTotal / T_TOTAL;
127   uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
128
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 124

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
123
124   uint256 private constant T_TOTAL = 1e17 * 10**DECIMALS;
125   uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
126   uint256 private _reflectionRate = _rTotal / T_TOTAL;
127   uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
128
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 125

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
124   uint256 private constant T_TOTAL = 1e17 * 10**DECIMALS;
125   uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
126   uint256 private _reflectionRate = _rTotal / T_TOTAL;
127   uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
128   uint256 private _tExcludedFromRewardTotal;
129
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 125

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
124   uint256 private constant T_TOTAL = 1e17 * 10**DECIMALS;
125   uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
126   uint256 private _reflectionRate = _rTotal / T_TOTAL;
127   uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
128   uint256 private _tExcludedFromRewardTotal;
129
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 126

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
125   uint256 private _rTotal = (MAX - (MAX % T_TOTAL));
126   uint256 private _reflectionRate = _rTotal / T_TOTAL;
127   uint256 private constant MIN_REFLECTION_RATE = T_TOTAL;
128   uint256 private _tExcludedFromRewardTotal;
129   uint256 private _rExcludedFromRewardTotal;
130
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
131
132   uint256 public antiWhaleAmt = 500_000_000_000_000 * 10**DECIMALS;
133   uint256 public swapTokensAtAmount = 50_000_000_000_000 * 10**DECIMALS;
134
135   // Anti Dump //
136
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
131
132   uint256 public antiWhaleAmt = 500_000_000_000_000 * 10**DECIMALS;
133   uint256 public swapTokensAtAmount = 50_000_000_000_000 * 10**DECIMALS;
134
135   // Anti Dump //
136
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 133

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
132   uint256 public antiWhaleAmt = 500_000_000_000_000 * 10**DECIMALS;
133   uint256 public swapTokensAtAmount = 50_000_000_000_000 * 10**DECIMALS;
134
135   // Anti Dump //
136   uint256 public maxSellAmountPerCycle = 15_000_000_000_000 * 10**DECIMALS;
137
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 133

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
132   uint256 public antiWhaleAmt = 500_000_000_000_000 * 10**DECIMALS;
133   uint256 public swapTokensAtAmount = 50_000_000_000_000 * 10**DECIMALS;
134
135   // Anti Dump //
136   uint256 public maxSellAmountPerCycle = 15_000_000_000_000 * 10**DECIMALS;
137
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 136

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
135   // Anti Dump //
136   uint256 public maxSellAmountPerCycle = 15_000_000_000_000 * 10**DECIMALS;
137   uint256 public antiDumpCycle = 1 hours;
138
139   // only allow Whitelist PancakeSwap Trading //
140
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 136

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
135   // Anti Dump //
136   uint256 public maxSellAmountPerCycle = 15_000_000_000_000 * 10**DECIMALS;
137   uint256 public antiDumpCycle = 1 hours;
138
139   // only allow Whitelist PancakeSwap Trading //
140
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 255

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
254   require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
255   _approve(sender, _msgSender(), currentAllowance - amount);
256   }
257
258   return true;
259
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 262

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
261   function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
262   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
263   return true;
264   }
265
266
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 269

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
268   require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
269   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
270
271   return true;
272   }
273
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 280

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
279   require(rAmount <= _rTotal, "Amount must be less than total reflections");
280   return rAmount / _reflectionRate;
281   }
282
283   //@dev kept original RFI naming -> "reward" as in reflection
284
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 291

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
290   _rOwned[account] = 0;
291   _tExcludedFromRewardTotal += tBalance;
292   _rExcludedFromRewardTotal += rBalance;
293   }
294   _isExcludedFromReward[account] = true;
295
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 292

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
291   _tExcludedFromRewardTotal += tBalance;
292   _rExcludedFromRewardTotal += rBalance;
293   }
294   _isExcludedFromReward[account] = true;
295   _excludedFromReward.push(account);
296
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 300

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
299   require(_isExcludedFromReward[account], "Account is not excluded");
300   for (uint256 i = 0; i < _excludedFromReward.length; i++) {
301   if (_excludedFromReward[i] == account) {
302   _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
303   uint256 tBalance = _tOwned[account];
304
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 302

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
301   if (_excludedFromReward[i] == account) {
302   _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
303   uint256 tBalance = _tOwned[account];
304   if (tBalance > 0) {
305   uint256 rBalance = tBalance * _reflectionRate;
306
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 305

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
304   if (tBalance > 0) {
305   uint256 rBalance = tBalance * _reflectionRate;
306   if (tBalance == _tExcludedFromRewardTotal) {
307   // try fix rTotal
308   if (rBalance > _rExcludedFromRewardTotal) {
309
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 309

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
308   if (rBalance > _rExcludedFromRewardTotal) {
309   _rTotal += rBalance - _rExcludedFromRewardTotal;
310   } else if (rBalance < _rExcludedFromRewardTotal) {
311   _rTotal -= _rExcludedFromRewardTotal - rBalance;
312   }
313
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 309

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
308   if (rBalance > _rExcludedFromRewardTotal) {
309   _rTotal += rBalance - _rExcludedFromRewardTotal;
310   } else if (rBalance < _rExcludedFromRewardTotal) {
311   _rTotal -= _rExcludedFromRewardTotal - rBalance;
312   }
313
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 311

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
310   } else if (rBalance < _rExcludedFromRewardTotal) {
311   _rTotal -= _rExcludedFromRewardTotal - rBalance;
312   }
313   _rExcludedFromRewardTotal = 0;
314
315
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 311

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
310   } else if (rBalance < _rExcludedFromRewardTotal) {
311   _rTotal -= _rExcludedFromRewardTotal - rBalance;
312   }
313   _rExcludedFromRewardTotal = 0;
314
315
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 316

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
315   } else {
316   _rExcludedFromRewardTotal -= rBalance;
317   }
318   _tOwned[account] = 0;
319   _rOwned[account] = rBalance;
320
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 320

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
319   _rOwned[account] = rBalance;
320   _tExcludedFromRewardTotal -= tBalance;
321   }
322   _isExcludedFromReward[account] = false;
323   _excludedFromReward.pop();
324
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 347

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
346
347   uint256 tSupply = T_TOTAL - _tExcludedFromRewardTotal;
348   uint256 rSupply = _rTotal - _rExcludedFromRewardTotal;
349   if (tSupply == 0) {
350   return;
351
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 348

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
347   uint256 tSupply = T_TOTAL - _tExcludedFromRewardTotal;
348   uint256 rSupply = _rTotal - _rExcludedFromRewardTotal;
349   if (tSupply == 0) {
350   return;
351   }
352
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 352

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
351   }
352   uint256 newRate = rSupply / tSupply;
353   if (newRate < MIN_REFLECTION_RATE) {
354   _reflectionRate = MIN_REFLECTION_RATE;
355   isRewardEnded = true;
356
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 404

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
403   if(transferType == ETransferType.Sell && !traderExcludedFromFee) {
404   bool newCycle = block.timestamp - userLastSell[from].lastSellTime >= antiDumpCycle;
405   if(!newCycle){
406   require(userLastSell[from].amountSoldInCycle + amount <= maxSellAmountPerCycle,
"You are exceeding maxSellAmountPerCycle");
407   userLastSell[from].amountSoldInCycle += amount;
408
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 406

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
405   if(!newCycle){
406   require(userLastSell[from].amountSoldInCycle + amount <= maxSellAmountPerCycle,
"You are exceeding maxSellAmountPerCycle");
407   userLastSell[from].amountSoldInCycle += amount;
408   }
409   else{
410
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
406   require(userLastSell[from].amountSoldInCycle + amount <= maxSellAmountPerCycle,
"You are exceeding maxSellAmountPerCycle");
407   userLastSell[from].amountSoldInCycle += amount;
408   }
409   else{
410   require(amount <= maxSellAmountPerCycle, "You are exceeding
maxSellAmountPerCycle");
411
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 423

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
422   traderExcludedFromFee ||
423   usedTaxes.rfi + usedTaxes.marketing + usedTaxes.liquidity + usedTaxes.burn == 0) {
424   taxFreeTransfer(from, to, amount);
425   } else {
426   _tokenTransfer(from, to, amount, usedTaxes);
427
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 423

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
422   traderExcludedFromFee ||
423   usedTaxes.rfi + usedTaxes.marketing + usedTaxes.liquidity + usedTaxes.burn == 0) {
424   taxFreeTransfer(from, to, amount);
425   } else {
426   _tokenTransfer(from, to, amount, usedTaxes);
427
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 423

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
422   traderExcludedFromFee ||
423   usedTaxes.rfi + usedTaxes.marketing + usedTaxes.liquidity + usedTaxes.burn == 0) {
424   taxFreeTransfer(from, to, amount);
425   } else {
426   _tokenTransfer(from, to, amount, usedTaxes);
427
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
437   if(usedTaxes.liquidity != 0) {
438   uint256 tLiquidity = tAmount * usedTaxes.liquidity / 100;
439   if (tLiquidity != 0) {
440   tTransferAmount -= tLiquidity;
441   totFeesPaid.liquidity += tLiquidity;
442
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
437   if(usedTaxes.liquidity != 0) {
438   uint256 tLiquidity = tAmount * usedTaxes.liquidity / 100;
439   if (tLiquidity != 0) {
440   tTransferAmount -= tLiquidity;
441   totFeesPaid.liquidity += tLiquidity;
442
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 440

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
439   if (tLiquidity != 0) {
440   tTransferAmount -= tLiquidity;
441   totFeesPaid.liquidity += tLiquidity;
442   _addBalance(address(this), tLiquidity, rate);
443   emit Transfer(sender, address(this), tLiquidity);
444
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 441

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
440   tTransferAmount -= tLiquidity;
441   totFeesPaid.liquidity += tLiquidity;
442   _addBalance(address(this), tLiquidity, rate);
443   emit Transfer(sender, address(this), tLiquidity);
444   }
445
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 447

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
446   if (usedTaxes.marketing != 0 && marketingAddress != DEAD_ADDRESS) {
447   uint256 tMarketing = tAmount * usedTaxes.marketing / 100;
448   if (tMarketing != 0) {
449   tTransferAmount -= tMarketing;
450   totFeesPaid.marketing += tMarketing;
451
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 447

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
446   if (usedTaxes.marketing != 0 && marketingAddress != DEAD_ADDRESS) {
447   uint256 tMarketing = tAmount * usedTaxes.marketing / 100;
448   if (tMarketing != 0) {
449   tTransferAmount -= tMarketing;
450   totFeesPaid.marketing += tMarketing;
451
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 449

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
448   if (tMarketing != 0) {
449   tTransferAmount -= tMarketing;
450   totFeesPaid.marketing += tMarketing;
451   _addBalance(marketingAddress, tMarketing, rate);
452   emit Transfer(sender, marketingAddress, tMarketing);
453
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 450

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
449   tTransferAmount -= tMarketing;
450   totFeesPaid.marketing += tMarketing;
451   _addBalance(marketingAddress, tMarketing, rate);
452   emit Transfer(sender, marketingAddress, tMarketing);
453   }
454
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 457

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
456   if (usedTaxes.burn != 0) {
457   uint256 tBurn = tAmount * usedTaxes.burn / 100;
458   if (tBurn != 0) {
459   tTransferAmount -= tBurn;
460   totFeesPaid.burn += tBurn;
461
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 457

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
456   if (usedTaxes.burn != 0) {
457   uint256 tBurn = tAmount * usedTaxes.burn / 100;
458   if (tBurn != 0) {
459   tTransferAmount -= tBurn;
460   totFeesPaid.burn += tBurn;
461
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 459

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
458   if (tBurn != 0) {
459   tTransferAmount -= tBurn;
460   totFeesPaid.burn += tBurn;
461   _addBalance(DEAD_ADDRESS, tBurn, rate);
462   emit Transfer(sender, DEAD_ADDRESS, tBurn);
463
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 460

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
459   tTransferAmount -= tBurn;
460   totFeesPaid.burn += tBurn;
461   _addBalance(DEAD_ADDRESS, tBurn, rate);
462   emit Transfer(sender, DEAD_ADDRESS, tBurn);
463   }
464
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 468

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
467   if (usedTaxes.rfi != 0 && !isRewardEnded) {
468   uint256 tRfi = tAmount * usedTaxes.rfi / 100;
469   if (tRfi != 0) {
470   tTransferAmount -= tRfi;
471   _rTotal -= tRfi * _reflectionRate;
472
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 468

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
467   if (usedTaxes.rfi != 0 && !isRewardEnded) {
468   uint256 tRfi = tAmount * usedTaxes.rfi / 100;
469   if (tRfi != 0) {
470   tTransferAmount -= tRfi;
471   _rTotal -= tRfi * _reflectionRate;
472
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 470

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
469   if (tRfi != 0) {
470   tTransferAmount -= tRfi;
471   _rTotal -= tRfi * _reflectionRate;
472   totFeesPaid.rfi += tRfi;
473   needRecalcReflectionRate = true;
474
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 471

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
470   tTransferAmount -= tRfi;
471   _rTotal -= tRfi * _reflectionRate;
472   totFeesPaid.rfi += tRfi;
473   needRecalcReflectionRate = true;
474   }
475
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 471

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
470   tTransferAmount -= tRfi;
471   _rTotal -= tRfi * _reflectionRate;
472   totFeesPaid.rfi += tRfi;
473   needRecalcReflectionRate = true;
474   }
475
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 472

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
471   _rTotal -= tRfi * _reflectionRate;
472   totFeesPaid.rfi += tRfi;
473   needRecalcReflectionRate = true;
474   }
475   }
476
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 490

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
489   //calculate how many tokens we need to exchange
490   uint256 tokensToSwap = contractTokenBalance / 2;
491   uint256 otherHalfOfTokens = tokensToSwap;
492   uint256 initialBalance = address(this).balance;
493   swapTokensForBNB(tokensToSwap, address(this));
494
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 494

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
493   swapTokensForBNB(tokensToSwap, address(this));
494   uint256 newBalance = address(this).balance - (initialBalance);
495   addLiquidity(otherHalfOfTokens, newBalance);
496   }
497
498
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 538

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
537   function updateAntiWhaleAmt(uint256 amount) external onlyOwner{
538   antiWhaleAmt = amount * 10**DECIMALS;
539   }
540
541   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
542
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 538

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
537   function updateAntiWhaleAmt(uint256 amount) external onlyOwner{
538   antiWhaleAmt = amount * 10**DECIMALS;
539   }
540
541   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
542
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 542

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
541   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
542   swapTokensAtAmount = amount * 10**DECIMALS;
543   }
544
545   function updateSwapEnabled(bool _enabled) external onlyOwner{
546
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 542

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
541   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
542   swapTokensAtAmount = amount * 10**DECIMALS;
543   }
544
545   function updateSwapEnabled(bool _enabled) external onlyOwner{
546
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 555

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
554   function bulkAntiBot(address[] memory accounts, bool state) external onlyOwner{
555   for(uint256 i = 0; i < accounts.length; i++){
556   _isBot[accounts[i]] = state;
557   }
558   }
559
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 565

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
564   function bulkPancakeSwapWhitelist(address[] memory accounts, bool state) external
onlyOwner{
565   for(uint256 i = 0; i < accounts.length; i++){
566   _isPancakeSwapWhitelisted[accounts[i]] = state;
567   }
568   }
569
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 577

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
576   require(_maxSellAmountPerCycle >= 1_000_000_000, "Amount must be >= 1B");
577   antiDumpCycle = timeInMinutes * 1 minutes;
578   maxSellAmountPerCycle = _maxSellAmountPerCycle * 10**DECIMALS;
579   }
580
581
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 578

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
577   antiDumpCycle = timeInMinutes * 1 minutes;
578   maxSellAmountPerCycle = _maxSellAmountPerCycle * 10**DECIMALS;
579   }
580
581   function isBot(address account) public view returns(bool){
582
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 578

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
577   antiDumpCycle = timeInMinutes * 1 minutes;
578   maxSellAmountPerCycle = _maxSellAmountPerCycle * 10**DECIMALS;
579   }
580
581   function isBot(address account) public view returns(bool){
582
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 595

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
594   if (_isExcludedFromReward[account]) {
595   _tOwned[account] += tAmount;
596   _tExcludedFromRewardTotal += tAmount;
597   _rExcludedFromRewardTotal += tAmount * rate;
598   } else {
599
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 596

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
595   _tOwned[account] += tAmount;
596   _tExcludedFromRewardTotal += tAmount;
597   _rExcludedFromRewardTotal += tAmount * rate;
598   } else {
599   _rOwned[account] += tAmount * rate;
600
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 597

low SEVERITY

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

Source File

- Catcoin.sol

Locations

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

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 597

low SEVERITY

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

Source File

- Catcoin.sol

Locations

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

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 599

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
598   } else {
599   _rOwned[account] += tAmount * rate;
600   }
601   }
602
603
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 599

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
598   } else {
599   _rOwned[account] += tAmount * rate;
600   }
601   }
602
603
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 605

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
604   if (_isExcludedFromReward[account]) {
605   _tOwned[account] -= tAmount;
606   _tExcludedFromRewardTotal -= tAmount;
607   _rExcludedFromRewardTotal -= tAmount * rate;
608   } else {
609
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 606

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
605   _tOwned[account] -= tAmount;
606   _tExcludedFromRewardTotal -= tAmount;
607   _rExcludedFromRewardTotal -= tAmount * rate;
608   } else {
609   _rOwned[account] -= tAmount * rate;
610
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 607

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
606   _tExcludedFromRewardTotal -= tAmount;
607   _rExcludedFromRewardTotal -= tAmount * rate;
608   } else {
609   _rOwned[account] -= tAmount * rate;
610   }
611
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 607

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
606   _tExcludedFromRewardTotal -= tAmount;
607   _rExcludedFromRewardTotal -= tAmount * rate;
608   } else {
609   _rOwned[account] -= tAmount * rate;
610   }
611
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 609

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
608   } else {
609   _rOwned[account] -= tAmount * rate;
610   }
611   }
612
613
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 609

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
608   } else {
609   _rOwned[account] -= tAmount * rate;
610   }
611   }
612
613
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 615

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
614   require(accounts.length == amounts.length, "Arrays must have the same size");
615   for(uint256 i= 0; i < accounts.length; i++){
616   taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);
617   }
618   }
619
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 616

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
615   for(uint256 i= 0; i < accounts.length; i++){
616   taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);
617   }
618   }
619
620
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 616

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
615   for(uint256 i= 0; i < accounts.length; i++){
616   taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);
617   }
618   }
619
620
```

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

LINE 302

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
301   if (_excludedFromReward[i] == account) {
302   _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
303   uint256 tBalance = _tOwned[account];
304   if (tBalance > 0) {
305   uint256 rBalance = tBalance * _reflectionRate;
306
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 10

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

- Catcoin.sol

Locations

```
9   // SPDX-License-Identifier: NOLICENSE
10   pragma solidity ^0.8.7;
11
12   interface IERC20 {
13   function totalSupply() external view returns (uint256);
14
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 301

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
300   for (uint256 i = 0; i < _excludedFromReward.length; i++) {
301   if (_excludedFromReward[i] == account) {
302   _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
303   uint256 tBalance = _tOwned[account];
304   if (tBalance > 0) {
305
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 302

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
301   if (_excludedFromReward[i] == account) {
302   _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
303   uint256 tBalance = _tOwned[account];
304   if (tBalance > 0) {
305   uint256 rBalance = tBalance * _reflectionRate;
306
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 302

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
301   if (_excludedFromReward[i] == account) {
302   _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
303   uint256 tBalance = _tOwned[account];
304   if (tBalance > 0) {
305   uint256 rBalance = tBalance * _reflectionRate;
306
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 516

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
515   address[] memory path = new address[](2);
516   path[0] = address(this);
517   path[1] = router.WETH();
518
519   _approve(address(this), address(router), tokenAmount);
520
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 517

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
516   path[0] = address(this);
517   path[1] = router.WETH();
518
519   _approve(address(this), address(router), tokenAmount);
520
521
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 556

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
555   for(uint256 i = 0; i < accounts.length; i++){
556   _isBot[accounts[i]] = state;
557   }
558   }
559
560
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 566

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
565   for(uint256 i = 0; i < accounts.length; i++){
566   _isPancakeSwapWhitelisted[accounts[i]] = state;
567   }
568   }
569
570
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 616

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
615   for(uint256 i= 0; i < accounts.length; i++){
616   taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);
617   }
618   }
619
620
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 616

low SEVERITY

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

Source File

- Catcoin.sol

Locations

```
615   for(uint256 i= 0; i < accounts.length; i++){
616   taxFreeTransfer(msg.sender, accounts[i], amounts[i] * 10**DECIMALS);
617   }
618   }
619
620
```

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.