CATCEO

Smart Contract Audit Report

SYSFIXED

03 Mar 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
CATCEO	CATCEO	Binance Smart Chain

Addresses

Contract address	0x4937e7d93dd8d8e76eb83659f109cdc633ffdee9
Contract deployer address	0xca94554ea45D90897F46F7d12aff87e4f2D997c1

Project Website

https://catceo.ai/

Codebase

https://bscscan.com/address/0x4937e7d93dd8d8e76eb83659f109cdc633ffdee9#code

SUMMARY

$CATCEO Tokens burning is used to increase the price of a crypto asset by controlling the supply. It involves sending crypto tokens to a wallet that does not have private keys. When tickets are sent to this wallet address, they essentially become inaccessible.

Contract Summary

Documentation Quality

CATCEO 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 CATCEO 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 132, 132, 132, 132, 133, 133, 135, 135, 232, 238, 248, 281, 296, 298, 320, 321, 326, 329, 331, 359, 359, 360, 360, 362, 362, 383, 389, 390, 392, 392, 400, 406, 409, 410, 412, 468, 472, 475, 476, 514 and 298.
- 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 297, 298, 298, 407, 407, 409, 410, 498, 499 and 515.

CONCLUSION

We have audited the CATCEO project released on March 2023 to discover issues and identify potential security vulnerabilities in CATCEO 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 CATCEO 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 some arithmetic operation issues, a floating pragma is set, and out-of-bounds array access which the index access expression can cause an exception in case an invalid array index value is used. The current pragma Solidity directive is ""^0.8.17"". Specifying a fixed compiler version is recommended 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	Thursday Mar 02 2023 04:33:01 GMT+0000 (Coordinated Universal Time)
Finished	Friday Mar 03 2023 03:36:07 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	CATCEO.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	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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
131
132   uint256 private _tTotal = 420 *10**15 * 10**_decimals;
133   uint256 private _rTotal = (MAX - (MAX % _tTotal));
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
131
132   uint256 private _tTotal = 420 *10**15 * 10**_decimals;
133   uint256 private _rTotal = (MAX - (MAX % _tTotal));
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
131
132   uint256 private _tTotal = 420 *10**15 * 10**_decimals;
133   uint256 private _rTotal = (MAX - (MAX % _tTotal));
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 132

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
131
132   uint256 private _tTotal = 420 *10**15 * 10**_decimals;
133   uint256 private _rTotal = (MAX - (MAX % _tTotal));
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 133

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
132   uint256 private _tTotal = 420 *10**15 * 10**_decimals;
133   uint256 private _rTotal = (MAX - (MAX % _tTotal));
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
137
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 133

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
132   uint256 private _tTotal = 420 *10**15 * 10**_decimals;
133   uint256 private _rTotal = (MAX - (MAX % _tTotal));
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
137
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 135

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
137   address public deadWallet = 0x000000000000000000000000000000000000dEaD;
138   address public marketingWallet = 0xca94554ea45D90897F46F7d12aff87e4f2D997c1;
139
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 135

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
134
135   uint256 public swapTokensAtAmount = 1e14 * 10**_decimals;
136
137   address public deadWallet = 0x000000000000000000000000000000000000dEaD;
138   address public marketingWallet = 0xca94554ea45D90897F46F7d12aff87e4f2D997c1;
139
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 232

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
231   require(currentAllowance >= amount, "BEP20: transfer amount exceeds allowance");
232   _approve(sender, _msgSender(), currentAllowance - amount);
233
234   return true;
235   }
236
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 238

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
237   function increaseAllowance(address spender, uint256 addedValue) public returns
(bool) {
238   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
239   return true;
240   }
241
242
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 248

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
247   require(currentAllowance >= subtractedValue, "BEP20: decreased allowance below
zero");
248   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
249
250   return true;
251   }
252
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 281

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
280   uint256 currentRate = _getRate();
281   return rAmount / currentRate;
282   }
283
284   //@dev kept original RFI naming -> "reward" as in reflection
285
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 296

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
295   require(_isExcluded[account], "Account is not excluded");
296   for (uint256 i = 0; i < _excluded.length; i++) {
297   if (_excluded[i] == account) {
298   _excluded[i] = _excluded[_excluded.length - 1];
299   _tOwned[account] = 0;
300
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 298

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
297   if (_excluded[i] == account) {
298   _excluded[i] = _excluded[_excluded.length - 1];
299   _tOwned[account] = 0;
300   _isExcluded[account] = false;
301   _excluded.pop();
302
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 320

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
319   function _reflectRfi(uint256 rRfi, uint256 tRfi) private {
320   _rTotal -= rRfi;
321   totFeesPaid.rfi += tRfi;
322   }
323
324
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 321

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
320   _rTotal -= rRfi;
321   totFeesPaid.rfi += tRfi;
322   }
323
324
325
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 326

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
325   function _takeMarketing(uint256 rMarketing, uint256 tMarketing) private {
326   totFeesPaid.marketing += tMarketing;
327
328   if (_isExcluded[address(this)]) {
329   _tOwned[address(this)] += tMarketing;
330
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 329

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
328   if (_isExcluded[address(this)]) {
329   _tOwned[address(this)] += tMarketing;
330   }
331   _rOwned[address(this)] += rMarketing;
332   }
333
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 331

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
330   }
331   _rOwned[address(this)] += rMarketing;
332   }
333
334
335
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 359

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
358
359   s.tRfi = (tAmount * taxes.rfi) / 100;
360   s.tMarketing = (tAmount * taxes.marketing) / 100;
361   s.tTransferAmount =
362   tAmount -
363
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 359

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
358
359   s.tRfi = (tAmount * taxes.rfi) / 100;
360   s.tMarketing = (tAmount * taxes.marketing) / 100;
361   s.tTransferAmount =
362   tAmount -
363
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 360

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
359   s.tRfi = (tAmount * taxes.rfi) / 100;
360   s.tMarketing = (tAmount * taxes.marketing) / 100;
361   s.tTransferAmount =
362   tAmount -
363   s.tRfi -
364
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 360

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
359   s.tRfi = (tAmount * taxes.rfi) / 100;
360   s.tMarketing = (tAmount * taxes.marketing) / 100;
361   s.tTransferAmount =
362   tAmount -
363   s.tRfi -
364
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 362

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
361   s.tTransferAmount =
362   tAmount -
363   s.tRfi -
364   s.tMarketing;
365   return s;
366
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 362

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
361   s.tTransferAmount =
362   tAmount -
363   s.tRfi -
364   s.tMarketing;
365   return s;
366
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 383

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
382   {
383   rAmount = tAmount * currentRate;
384
385   if (!takeFee) {
386   return (rAmount, rAmount, 0, 0);
387
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 389

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
388
389   rRfi = s.tRfi * currentRate;
390   rMarketing = s.tMarketing * currentRate;
391   rTransferAmount =
392   rAmount -
393
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 390

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
389   rRfi = s.tRfi * currentRate;
390   rMarketing = s.tMarketing * currentRate;
391   rTransferAmount =
392   rAmount -
393   rRfi -
394
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 392

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
391   rTransferAmount =
392   rAmount -
393   rRfi -
394   rMarketing;
395   return (rAmount, rTransferAmount, rRfi, rMarketing);
396
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 392

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
391   rTransferAmount =
392   rAmount -
393   rRfi -
394   rMarketing;
395   return (rAmount, rTransferAmount, rRfi, rMarketing);
396
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 400

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
399   (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
400   return rSupply / tSupply;
401   }
402
403   function _getCurrentSupply() private view returns (uint256, uint256) {
404
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 406

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
405   uint256 tSupply = _tTotal;
406   for (uint256 i = 0; i < _excluded.length; i++) {
407   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply)
408   return (_rTotal, _tTotal);
409   rSupply = rSupply - _rOwned[_excluded[i]];
410
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 409

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
408   return (_rTotal, _tTotal);
409   rSupply = rSupply - _rOwned[_excluded[i]];
410   tSupply = tSupply - _tOwned[_excluded[i]];
411   }
412   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
413
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 410

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
409   rSupply = rSupply - _rOwned[_excluded[i]];
410   tSupply = tSupply - _tOwned[_excluded[i]];
411   }
412   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
413   return (rSupply, tSupply);
414
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 412

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
411   }
412   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
413   return (rSupply, tSupply);
414   }
415
416
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 468

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
467   //from excluded
468   _tOwned[sender] = _tOwned[sender] - tAmount;
469   }
470   if (_isExcluded[recipient]) {
471   //to excluded
472
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 472

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
471   //to excluded
472   _tOwned[recipient] = _tOwned[recipient] + s.tTransferAmount;
473   }
474
475   _rOwned[sender] = _rOwned[sender] - s.rAmount;
476
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 475

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
474
475   _rOwned[sender] = _rOwned[sender] - s.rAmount;
476   _rOwned[recipient] = _rOwned[recipient] + s.rTransferAmount;
477
478   if (s.rRfi > 0 || s.tRfi > 0) _reflectRfi(s.rRfi, s.tRfi);
479
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 476

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
475   _rOwned[sender] = _rOwned[sender] - s.rAmount;
476   _rOwned[recipient] = _rOwned[recipient] + s.rTransferAmount;
477
478   if (s.rRfi > 0 || s.tRfi > 0) _reflectRfi(s.rRfi, s.tRfi);
479   if (s.rMarketing > 0 || s.tMarketing > 0) _takeMarketing(s.rMarketing,
s.tMarketing);
480
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 514

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
513   function bulkExcludeFee(address[] memory accounts, bool state) external onlyOwner {
514   for (uint256 i = 0; i < accounts.length; i++) {
515   _isExcludedFromFee[accounts[i]] = state;
516   }
517   }
518
```

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

LINE 298

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
297   if (_excluded[i] == account) {
298   _excluded[i] = _excluded[_excluded.length - 1];
299   _tOwned[account] = 0;
300   _isExcluded[account] = false;
301   _excluded.pop();
302
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

low SEVERITY

The current pragma Solidity directive is ""^0.8.17"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source File

- CATCEO.sol

Locations

```
5   // SPDX-License-Identifier: UNLICENSE
6   pragma solidity ^0.8.17;
7
8   interface IBEP20 {
9   function totalSupply() external view returns (uint256);
10
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 297

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
296   for (uint256 i = 0; i < _excluded.length; i++) {
297   if (_excluded[i] == account) {
298   _excluded[i] = _excluded[_excluded.length - 1];
299   _tOwned[account] = 0;
300   _isExcluded[account] = false;
301
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 298

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
297   if (_excluded[i] == account) {
298   _excluded[i] = _excluded[_excluded.length - 1];
299   _tOwned[account] = 0;
300   _isExcluded[account] = false;
301   _excluded.pop();
302
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 298

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
297   if (_excluded[i] == account) {
298   _excluded[i] = _excluded[_excluded.length - 1];
299   _tOwned[account] = 0;
300   _isExcluded[account] = false;
301   _excluded.pop();
302
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 407

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
406   for (uint256 i = 0; i < _excluded.length; i++) {
407   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply)
408   return (_rTotal, _tTotal);
409   rSupply = rSupply - _rOwned[_excluded[i]];
410   tSupply = tSupply - _tOwned[_excluded[i]];
411
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 407

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
406   for (uint256 i = 0; i < _excluded.length; i++) {
407   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply)
408   return (_rTotal, _tTotal);
409   rSupply = rSupply - _rOwned[_excluded[i]];
410   tSupply = tSupply - _tOwned[_excluded[i]];
411
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 409

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
408   return (_rTotal, _tTotal);
409   rSupply = rSupply - _rOwned[_excluded[i]];
410   tSupply = tSupply - _tOwned[_excluded[i]];
411   }
412   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
413
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 410

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
409   rSupply = rSupply - _rOwned[_excluded[i]];
410   tSupply = tSupply - _tOwned[_excluded[i]];
411   }
412   if (rSupply < _rTotal / _tTotal) return (_rTotal, _tTotal);
413   return (rSupply, tSupply);
414
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 498

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
497   address[] memory path = new address[](2);
498   path[0] = address(this);
499   path[1] = router.WETH();
500
501   _approve(address(this), address(router), tokenAmount);
502
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 499

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
498   path[0] = address(this);
499   path[1] = router.WETH();
500
501   _approve(address(this), address(router), tokenAmount);
502
503
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 515

low SEVERITY

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

Source File

- CATCEO.sol

Locations

```
514   for (uint256 i = 0; i < accounts.length; i++) {
515   _isExcludedFromFee[accounts[i]] = state;
516   }
517   }
518
519
```

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.