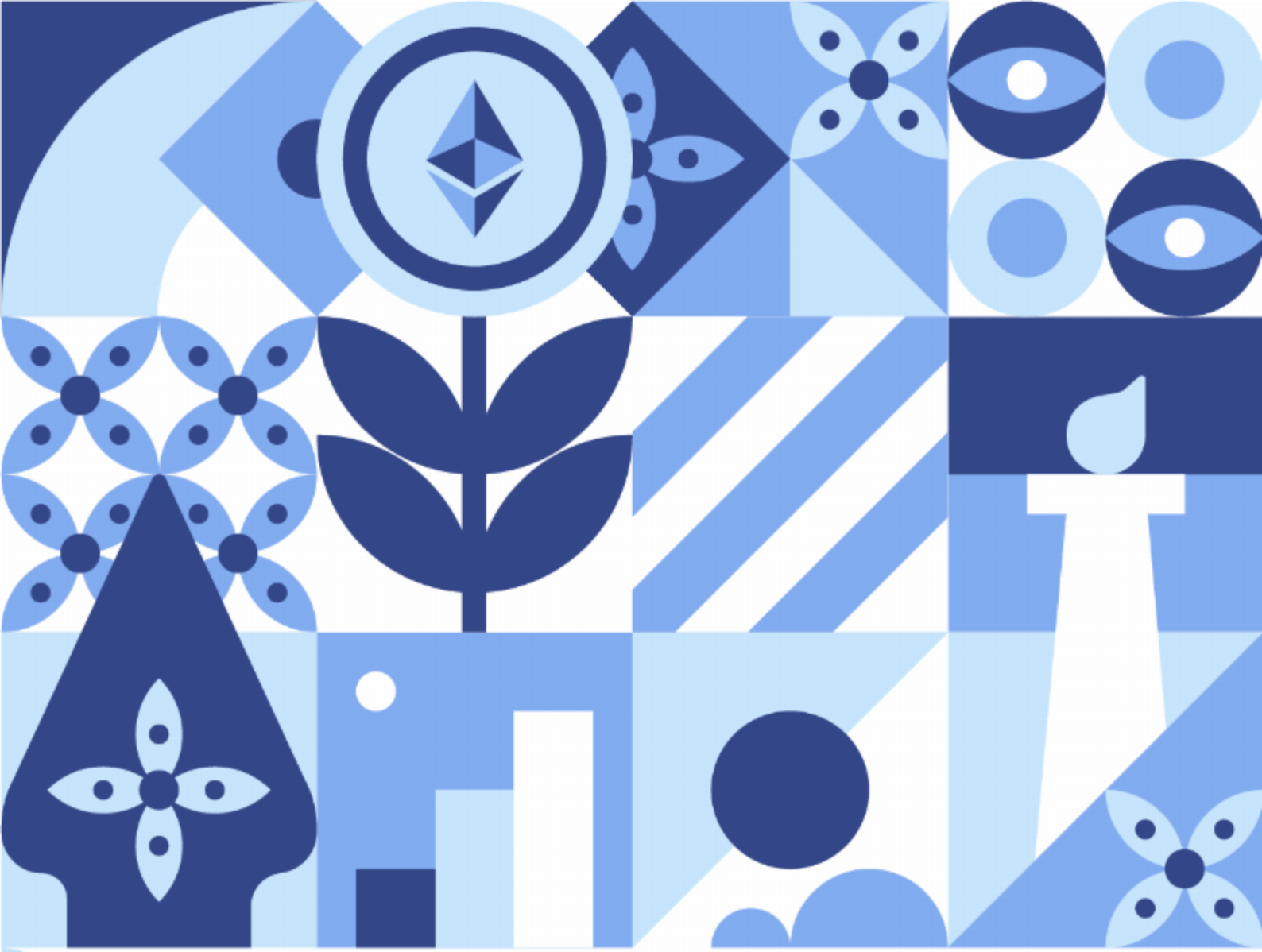

Medic Inu

Smart Contract Audit Report

SYSFIXED

28 Feb 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Medic Inu	MEDS	Ethereum

Addresses

Contract address	0xcabc94c68CC395a8e7b4CDBF29d89c5DF302f416
Contract deployer address	0xe7C17AE03DAdca2071cF12923A6bbC3841e65dE5

Project Website

https://www.medicinu.xyz/

Codebase

https://etherscan.io/address/0xcabc94c68CC395a8e7b4CDBF29d89c5DF302f416#code

SUMMARY

First inspired by the war in Ukraine, Medic Inu was created to help the world heal. He is sent on missions around the world to help restore the lives of his fellow canines and people affected by tragedy. He is backed by an audited protocol and a KYC'd team. 2%+ of every transaction is being donated to various non-profit charities.

Contract Summary

Documentation Quality

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

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

Code Quality

The Overall quality of the basecode is standard.

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

Test Coverage

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

Audit Findings Summary

- SWC-100 SWC-108 | Explicitly define visibility for all state variables on lines 953.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 111, 147, 170, 171, 210, 250, 522, 910, 910, 911, 911, 956, 956, 957, 957, 958, 958, 1184, 1187, 1207, 1209, 1241, 1241, 1241, 1241, 1256, 1256, 1256, 1256, 1277, 1277, 1277, 1277, 1278, 1278, 1278, 1278, 1298, 1306, 1335, 1382, 1421, 1429, 1437, 1445, 1449, 1546, 1559, 1559, 1559, 1559, 1559, 1559, 1565, 1565, 1566, 1567, 1573, 1574, 1575, 1575, 1576, 1577, 1585, 1585, 1587, 1588, 1588, 1588, 1591, 1591, 1591, 1187 and 1209.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 2.
- 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 1185, 1186, 1186, 1208, 1209, 1209, 1384, 1385, 1387, 1388, 1605 and 1606.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 1506.

CONCLUSION

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

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

The issues found in the Medic Inu 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 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 of the use of an invalid array index value.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	ISSUE FOUND
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.	ISSUE FOUND
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	Sunday Feb 27 2022 16:15:16 GMT+0000 (Coordinated Universal Time)
Finished	Monday Feb 28 2022 16:27:52 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	MedicInu.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	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-101	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-115	USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.	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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 111

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
110   function add(uint256 a, uint256 b) internal pure returns (uint256) {
111   uint256 c = a + b;
112   require(c >= a, "SafeMath: addition overflow");
113
114   return c;
115
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 147

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
146   require(b <= a, errorMessage);
147   uint256 c = a - b;
148
149   return c;
150   }
151
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 170

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
169
170   uint256 c = a * b;
171   require(c / a == b, "SafeMath: multiplication overflow");
172
173   return c;
174
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 171

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
170   uint256 c = a * b;
171   require(c / a == b, "SafeMath: multiplication overflow");
172
173   return c;
174   }
175
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 210

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
209   require(b > 0, errorMessage);
210   uint256 c = a / b;
211   // assert(a == b * c + a % b); // There is no case in which this doesn't hold
212
213   return c;
214
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 250

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
249   require(b != 0, errorMessage);
250   return a % b;
251   }
252   }
253
254
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 522

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
521   _owner = address(0);
522   _lockTime = block.timestamp + time;
523   emit OwnershipTransferred(_owner, address(0));
524   }
525
526
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 910

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
909   uint256 private constant MAX = ~uint256(0);
910   uint256 private _tTotal = 1_000_000_000 * 10**9;
911   uint256 private _rTotal = (MAX - (MAX % _tTotal));
912   uint256 private _tFeeTotal;
913
914
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 910

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
909   uint256 private constant MAX = ~uint256(0);
910   uint256 private _tTotal = 1_000_000_000 * 10**9;
911   uint256 private _rTotal = (MAX - (MAX % _tTotal));
912   uint256 private _tFeeTotal;
913
914
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 911

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
910   uint256 private _tTotal = 1_000_000_000 * 10**9;
911   uint256 private _rTotal = (MAX - (MAX % _tTotal));
912   uint256 private _tFeeTotal;
913
914   address payable public _marketingAddress =
915
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 911

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
910   uint256 private _tTotal = 1_000_000_000 * 10**9;
911   uint256 private _rTotal = (MAX - (MAX % _tTotal));
912   uint256 private _tFeeTotal;
913
914   address payable public _marketingAddress =
915
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 956

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
955
956   uint256 public _maxTxAmount = 10_000_000 * 10**_decimals;
957   uint256 private numTokensSellToAddToLiquidity = 6_000_000 * 10**_decimals;
958   uint256 public _maxWalletSize = 10_000_000 * 10**_decimals;
959
960
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 956

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
955
956   uint256 public _maxTxAmount = 10_000_000 * 10**_decimals;
957   uint256 private numTokensSellToAddToLiquidity = 6_000_000 * 10**_decimals;
958   uint256 public _maxWalletSize = 10_000_000 * 10**_decimals;
959
960
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 957

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
956   uint256 public _maxTxAmount = 10_000_000 * 10**_decimals;
957   uint256 private numTokensSellToAddToLiquidity = 6_000_000 * 10**_decimals;
958   uint256 public _maxWalletSize = 10_000_000 * 10**_decimals;
959
960   event SwapAndLiquifyEnabledUpdated(bool enabled);
961
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 957

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
956   uint256 public _maxTxAmount = 10_000_000 * 10**_decimals;
957   uint256 private numTokensSellToAddToLiquidity = 6_000_000 * 10**_decimals;
958   uint256 public _maxWalletSize = 10_000_000 * 10**_decimals;
959
960   event SwapAndLiquifyEnabledUpdated(bool enabled);
961
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 958

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
957   uint256 private numTokensSellToAddToLiquidity = 6_000_000 * 10**_decimals;
958   uint256 public _maxWalletSize = 10_000_000 * 10**_decimals;
959
960   event SwapAndLiquifyEnabledUpdated(bool enabled);
961   event SwapAndLiquify(
962
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 958

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
957   uint256 private numTokensSellToAddToLiquidity = 6_000_000 * 10**_decimals;
958   uint256 public _maxWalletSize = 10_000_000 * 10**_decimals;
959
960   event SwapAndLiquifyEnabledUpdated(bool enabled);
961   event SwapAndLiquify(
962
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1184

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1183   require(_isBlackListedBot[account], "Account is not blacklisted");
1184   for (uint256 i = 0; i < _blackListedBots.length; i++) {
1185   if (_blackListedBots[i] == account) {
1186   _blackListedBots[i] = _blackListedBots[
1187   _blackListedBots.length - 1
1188
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1187

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1186   _blackListedBots[i] = _blackListedBots[
1187   _blackListedBots.length - 1
1188   ];
1189   _isBlackListedBot[account] = false;
1190   _blackListedBots.pop();
1191
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1207

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1206   require(_isExcluded[account], "Account is not excluded");
1207   for (uint256 i = 0; i < _excluded.length; i++) {
1208   if (_excluded[i] == account) {
1209   _excluded[i] = _excluded[_excluded.length - 1];
1210   _tOwned[account] = 0;
1211
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1209

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1208   if (_excluded[i] == account) {
1209   _excluded[i] = _excluded[_excluded.length - 1];
1210   _tOwned[account] = 0;
1211   _isExcluded[account] = false;
1212   _excluded.pop();
1213
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1241

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1240   ) external onlyOwner {
1241   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1242   sellFee.tax = tax;
1243   sellFee.marketing = marketing;
1244   sellFee.liquidity = liquidity;
1245
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1241

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1240   ) external onlyOwner {
1241   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1242   sellFee.tax = tax;
1243   sellFee.marketing = marketing;
1244   sellFee.liquidity = liquidity;
1245
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1241

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1240   ) external onlyOwner {
1241   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1242   sellFee.tax = tax;
1243   sellFee.marketing = marketing;
1244   sellFee.liquidity = liquidity;
1245
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1241

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1240   ) external onlyOwner {
1241   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1242   sellFee.tax = tax;
1243   sellFee.marketing = marketing;
1244   sellFee.liquidity = liquidity;
1245
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1256

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1255   ) external onlyOwner {
1256   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1257   buyFee.tax = tax;
1258   buyFee.marketing = marketing;
1259   buyFee.liquidity = liquidity;
1260
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1256

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1255   ) external onlyOwner {
1256   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1257   buyFee.tax = tax;
1258   buyFee.marketing = marketing;
1259   buyFee.liquidity = liquidity;
1260
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1256

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1255   ) external onlyOwner {
1256   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1257   buyFee.tax = tax;
1258   buyFee.marketing = marketing;
1259   buyFee.liquidity = liquidity;
1260
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1256

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1255   ) external onlyOwner {
1256   require((tax + liquidity + marketing + dev + donation) <= 11, "Tax much");
1257   buyFee.tax = tax;
1258   buyFee.marketing = marketing;
1259   buyFee.liquidity = liquidity;
1260
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1277

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1276   ) external onlyOwner {
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1277

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1276   ) external onlyOwner {
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1277

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1276   ) external onlyOwner {
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1277

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1276   ) external onlyOwner {
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1278

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281   buyFee.liquidity = buy_liquidity;
1282
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1278

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281   buyFee.liquidity = buy_liquidity;
1282
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1278

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281   buyFee.liquidity = buy_liquidity;
1282
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1278

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1277   require((buy_tax + buy_liquidity + buy_marketing + buy_dev + buy_donation) <= 11,
"Tax much");
1278   require((sell_tax + sell_liquidity + sell_marketing + sell_dev + sell_donation) <=
11, "Tax much");
1279   buyFee.tax = buy_tax;
1280   buyFee.marketing = buy_marketing;
1281   buyFee.liquidity = buy_liquidity;
1282
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1298

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1297   require(maxTxPercent >= 3, "MaxTx little"); // maxTx can never be below 0.3% of
total supply
1298   _maxTxAmount = _tTotal.mul(maxTxPercent).div(10**3);
1299   }
1300
1301   function _setMaxWalletSizePercent(uint256 maxWalletSize)
1302
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1306

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1305   require(maxWalletSize >= 10, "MaxWallet little"); // maxWallet can never be below
1% of total supply
1306   _maxWalletSize = _tTotal.mul(maxWalletSize).div(10**3);
1307   }
1308
1309   function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
1310
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1335

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1334   uint256 tLiquidity = calculateLiquidityFee(tAmount);
1335   uint256 tWallet = calculateMarketingFee(tAmount) +
1336   calculateDevFee(tAmount);
1337   uint256 tDonation = calculateDonationFee(tAmount);
1338   uint256 tTransferAmount = tAmount.sub(tFee).sub(tLiquidity);
1339
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1382

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1381   uint256 tSupply = _tTotal;
1382   for (uint256 i = 0; i < _excluded.length; i++) {
1383   if (
1384   _rOwned[_excluded[i]] > rSupply ||
1385   _tOwned[_excluded[i]] > tSupply
1386
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1421

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1420   function calculateTaxFee(uint256 _amount) private view returns (uint256) {
1421   return _amount.mul(_taxFee).div(10**2);
1422   }
1423
1424   function calculateLiquidityFee(uint256 _amount)
1425
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1429

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1428   {
1429   return _amount.mul(_liquidityFee).div(10**2);
1430   }
1431
1432   function calculateMarketingFee(uint256 _amount)
1433
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1437

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1436   {
1437   return _amount.mul(_marketingFee).div(10**2);
1438   }
1439
1440   function calculateDonationFee(uint256 _amount)
1441
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1445

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1444   {
1445   return _amount.mul(_donationFee).div(10**2);
1446   }
1447
1448   function calculateDevFee(uint256 _amount) private view returns (uint256) {
1449
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1449

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1448   function calculateDevFee(uint256 _amount) private view returns (uint256) {
1449   return _amount.mul(_devFee).div(10**2);
1450   }
1451
1452   function removeAllFee() private {
1453
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1546

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1545   require(
1546   amount + balanceOf(to) <= _maxWalletSize,
1547   "Recipient exceeds max wallet size."
1548   );
1549   }
1550
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1559

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1558   // Split the contract balance into halves
1559   uint256 denominator = (buyFee.liquidity +
1560   sellFee.liquidity +
1561   buyFee.marketing +
1562   sellFee.marketing +
1563
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1559

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1558   // Split the contract balance into halves
1559   uint256 denominator = (buyFee.liquidity +
1560   sellFee.liquidity +
1561   buyFee.marketing +
1562   sellFee.marketing +
1563
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1559

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1558   // Split the contract balance into halves
1559   uint256 denominator = (buyFee.liquidity +
1560   sellFee.liquidity +
1561   buyFee.marketing +
1562   sellFee.marketing +
1563
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1559

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1558   // Split the contract balance into halves
1559   uint256 denominator = (buyFee.liquidity +
1560   sellFee.liquidity +
1561   buyFee.marketing +
1562   sellFee.marketing +
1563
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1559

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1558   // Split the contract balance into halves
1559   uint256 denominator = (buyFee.liquidity +
1560   sellFee.liquidity +
1561   buyFee.marketing +
1562   sellFee.marketing +
1563
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1559

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1558   // Split the contract balance into halves
1559   uint256 denominator = (buyFee.liquidity +
1560   sellFee.liquidity +
1561   buyFee.marketing +
1562   sellFee.marketing +
1563
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1565

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1564   sellFee.dev) * 2;
1565   uint256 tokensToAddLiquidityWith = (tokens *
1566   (buyFee.liquidity + sellFee.liquidity)) / denominator;
1567   uint256 toSwap = tokens - tokensToAddLiquidityWith;
1568
1569
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1565

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1564   sellFee.dev) * 2;
1565   uint256 tokensToAddLiquidityWith = (tokens *
1566   (buyFee.liquidity + sellFee.liquidity)) / denominator;
1567   uint256 toSwap = tokens - tokensToAddLiquidityWith;
1568
1569
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1566

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1565   uint256 tokensToAddLiquidityWith = (tokens *
1566   (buyFee.liquidity + sellFee.liquidity)) / denominator;
1567   uint256 toSwap = tokens - tokensToAddLiquidityWith;
1568
1569   uint256 initialBalance = address(this).balance;
1570
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1567

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1566   (buyFee.liquidity + sellFee.liquidity)) / denominator;
1567   uint256 toSwap = tokens - tokensToAddLiquidityWith;
1568
1569   uint256 initialBalance = address(this).balance;
1570
1571
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1573

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1572
1573   uint256 deltaBalance = address(this).balance - initialBalance;
1574   uint256 unitBalance = deltaBalance /
1575   (denominator - (buyFee.liquidity + sellFee.liquidity));
1576   uint256 bnbToAddLiquidityWith = unitBalance *
1577
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1574

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1573   uint256 deltaBalance = address(this).balance - initialBalance;
1574   uint256 unitBalance = deltaBalance /
1575   (denominator - (buyFee.liquidity + sellFee.liquidity));
1576   uint256 bnbToAddLiquidityWith = unitBalance *
1577   (buyFee.liquidity + sellFee.liquidity);
1578
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1575

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1574   uint256 unitBalance = deltaBalance /
1575   (denominator - (buyFee.liquidity + sellFee.liquidity));
1576   uint256 bnbToAddLiquidityWith = unitBalance *
1577   (buyFee.liquidity + sellFee.liquidity);
1578
1579
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1575

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1574   uint256 unitBalance = deltaBalance /
1575   (denominator - (buyFee.liquidity + sellFee.liquidity));
1576   uint256 bnbToAddLiquidityWith = unitBalance *
1577   (buyFee.liquidity + sellFee.liquidity);
1578
1579
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1576

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1575   (denominator - (buyFee.liquidity + sellFee.liquidity));
1576   uint256 bnbToAddLiquidityWith = unitBalance *
1577   (buyFee.liquidity + sellFee.liquidity);
1578
1579   if (bnbToAddLiquidityWith > 0) {
1580
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1577

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1576   uint256 bnbToAddLiquidityWith = unitBalance *
1577   (buyFee.liquidity + sellFee.liquidity);
1578
1579   if (bnbToAddLiquidityWith > 0) {
1580   // Add liquidity to pancake
1581
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1585

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1584   // Send ETH to marketing
1585   uint256 marketingAmt = unitBalance *
1586   2 *
1587   (buyFee.marketing + sellFee.marketing);
1588   uint256 devAmt = unitBalance * 2 * (buyFee.dev + sellFee.dev) >
1589
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1585

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1584   // Send ETH to marketing
1585   uint256 marketingAmt = unitBalance *
1586   2 *
1587   (buyFee.marketing + sellFee.marketing);
1588   uint256 devAmt = unitBalance * 2 * (buyFee.dev + sellFee.dev) >
1589
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1587

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1586   2 *
1587   (buyFee.marketing + sellFee.marketing);
1588   uint256 devAmt = unitBalance * 2 * (buyFee.dev + sellFee.dev) >
1589   address(this).balance
1590   ? address(this).balance
1591
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1588

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1587   (buyFee.marketing + sellFee.marketing);
1588   uint256 devAmt = unitBalance * 2 * (buyFee.dev + sellFee.dev) >
1589   address(this).balance
1590   ? address(this).balance
1591   : unitBalance * 2 * (buyFee.dev + sellFee.dev);
1592
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1588

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1587   (buyFee.marketing + sellFee.marketing);
1588   uint256 devAmt = unitBalance * 2 * (buyFee.dev + sellFee.dev) >
1589   address(this).balance
1590   ? address(this).balance
1591   : unitBalance * 2 * (buyFee.dev + sellFee.dev);
1592
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1588

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1587   (buyFee.marketing + sellFee.marketing);
1588   uint256 devAmt = unitBalance * 2 * (buyFee.dev + sellFee.dev) >
1589   address(this).balance
1590   ? address(this).balance
1591   : unitBalance * 2 * (buyFee.dev + sellFee.dev);
1592
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1591

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1590   ? address(this).balance
1591   : unitBalance * 2 * (buyFee.dev + sellFee.dev);
1592
1593   if (marketingAmt > 0) {
1594   payable(_marketingAddress).transfer(marketingAmt);
1595
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1591

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1590   ? address(this).balance
1591   : unitBalance * 2 * (buyFee.dev + sellFee.dev);
1592
1593   if (marketingAmt > 0) {
1594   payable(_marketingAddress).transfer(marketingAmt);
1595
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1591

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1590   ? address(this).balance
1591   : unitBalance * 2 * (buyFee.dev + sellFee.dev);
1592
1593   if (marketingAmt > 0) {
1594   payable(_marketingAddress).transfer(marketingAmt);
1595
```

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

LINE 1187

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1186   _blackListedBots[i] = _blackListedBots[
1187   _blackListedBots.length - 1
1188   ];
1189   _isBlackListedBot[account] = false;
1190   _blackListedBots.pop();
1191
```

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

LINE 1209

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1208   if (_excluded[i] == account) {
1209   _excluded[i] = _excluded[_excluded.length - 1];
1210   _tOwned[account] = 0;
1211   _isExcluded[account] = false;
1212   _excluded.pop();
1213
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 2

low SEVERITY

The current pragma Solidity directive is ""^0.8.10"". 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

- MedicInu.sol

Locations

```
1   // SPDX-License-Identifier: Unlicensed
2   pragma solidity ^0.8.10;
3
4   interface IERC20 {
5   function totalSupply() external view returns (uint256);
6
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 953

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
952
953   bool inSwapAndLiquify;
954   bool public swapAndLiquifyEnabled = true;
955
956   uint256 public _maxTxAmount = 10_000_000 * 10**_decimals;
957
```

SWC-115 | USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.

LINE 1506

low SEVERITY

Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" unless you really know what you are doing.

Source File

- MedicInu.sol

Locations

```
1505   require(!_isBlackListedBot[msg.sender], "blacklisted");
1506   require(!_isBlackListedBot[tx.origin], "blacklisted");
1507
1508   // is the token balance of this contract address over the min number of
1509   // tokens that we need to initiate a swap + liquidity lock?
1510
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1185

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1184   for (uint256 i = 0; i < _blackListedBots.length; i++) {
1185   if (_blackListedBots[i] == account) {
1186   _blackListedBots[i] = _blackListedBots[
1187   _blackListedBots.length - 1
1188   ];
1189
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1186

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1185   if (_blackListedBots[i] == account) {
1186   _blackListedBots[i] = _blackListedBots[
1187   _blackListedBots.length - 1
1188   ];
1189   _isBlackListedBot[account] = false;
1190
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1186

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1185   if (_blackListedBots[i] == account) {
1186   _blackListedBots[i] = _blackListedBots[
1187   _blackListedBots.length - 1
1188   ];
1189   _isBlackListedBot[account] = false;
1190
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1208

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1207   for (uint256 i = 0; i < _excluded.length; i++) {
1208   if (_excluded[i] == account) {
1209   _excluded[i] = _excluded[_excluded.length - 1];
1210   _tOwned[account] = 0;
1211   _isExcluded[account] = false;
1212
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1209

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1208   if (_excluded[i] == account) {
1209   _excluded[i] = _excluded[_excluded.length - 1];
1210   _tOwned[account] = 0;
1211   _isExcluded[account] = false;
1212   _excluded.pop();
1213
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1209

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1208   if (_excluded[i] == account) {
1209   _excluded[i] = _excluded[_excluded.length - 1];
1210   _tOwned[account] = 0;
1211   _isExcluded[account] = false;
1212   _excluded.pop();
1213
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1384

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1383   if (
1384   _rOwned[_excluded[i]] > rSupply ||
1385   _tOwned[_excluded[i]] > tSupply
1386   ) return (_rTotal, _tTotal);
1387   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1388
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1385

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1384   _rOwned[_excluded[i]] > rSupply ||
1385   _tOwned[_excluded[i]] > tSupply
1386   ) return (_rTotal, _tTotal);
1387   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1388   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1389
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1387

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1386   ) return (_rTotal, _tTotal);
1387   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1388   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1389   }
1390   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
1391
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1388

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1387   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
1388   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
1389   }
1390   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
1391   return (rSupply, tSupply);
1392
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1605

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1604   address[] memory path = new address[](2);
1605   path[0] = address(this);
1606   path[1] = uniswapV2Router.WETH();
1607
1608   _approve(address(this), address(uniswapV2Router), tokenAmount);
1609
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1606

low SEVERITY

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

Source File

- MedicInu.sol

Locations

```
1605   path[0] = address(this);
1606   path[1] = uniswapV2Router.WETH();
1607
1608   _approve(address(this), address(uniswapV2Router), tokenAmount);
1609
1610
```

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.