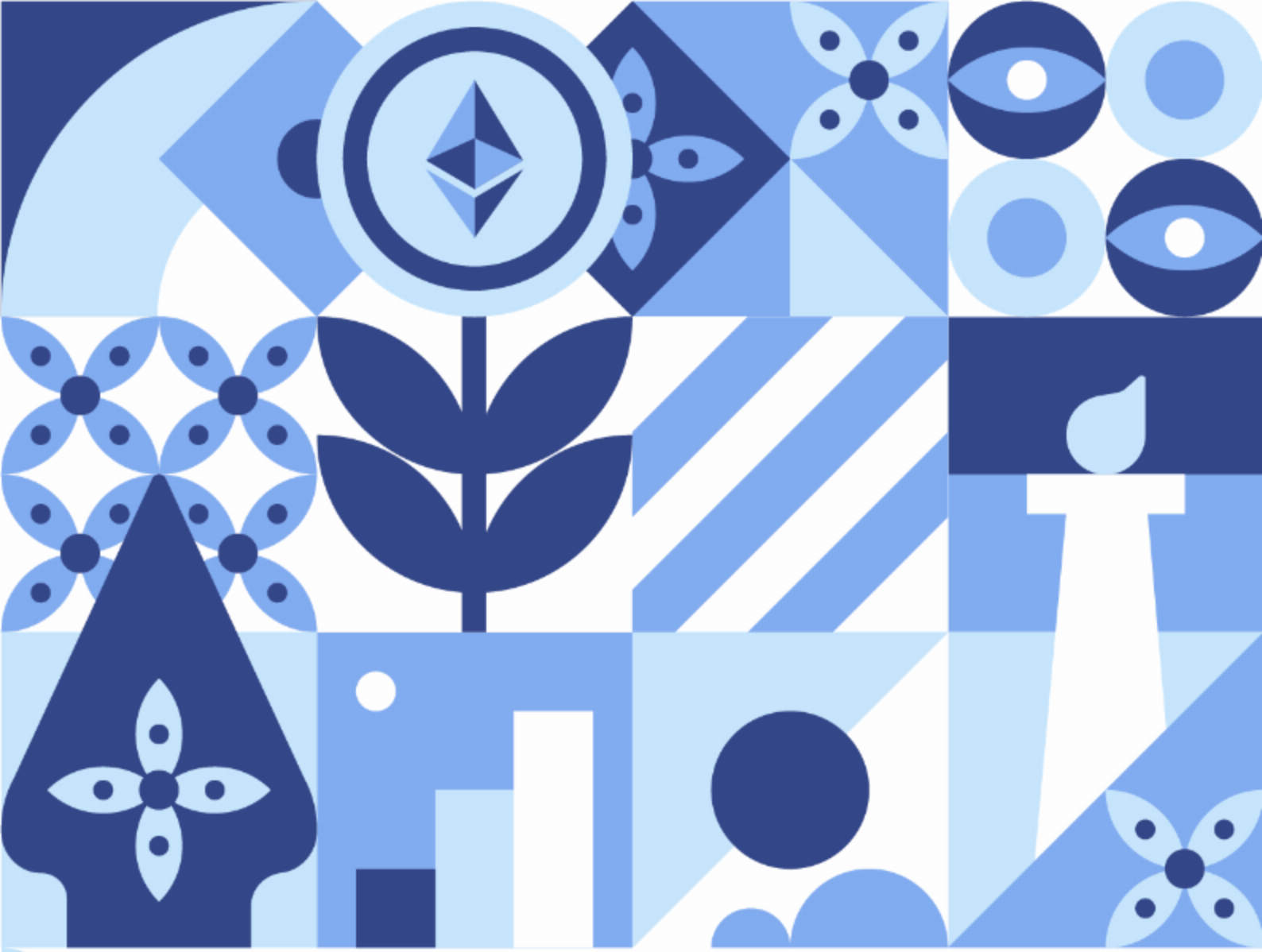

MASYA

Smart Contract Audit Report

SYSFIXED

25 Oct 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
MASYA	MASYA	Ethereum

Addresses

Contract address	0x26f45C6D6bfdd89d37a8856838c2141348334E0F
Contract deployer address	0x0cefB0Fb76a7d2081c1905057aab2D21e07E4beB

Project Website

https://www.masya.io/

Codebase

https://etherscan.io/address/0x26f45C6D6bfdd89d37a8856838c2141348334E0F#code

SUMMARY

$MASYA Is A Decentralised, Zero Tax Meme Coin On The Ethereum Network. Dedicated To Vitalik Buterin And Cats Everywhere.

Contract Summary

Documentation Quality

MASYA 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 MASYA 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 386, 405, 427, 460, 462, 483, 484, 509, 511, 607, 621, 636, 637, 650, 662, 677, 691, 705, 719, 735, 758, 781, 807, 1141, 1143, 1144, 1145, 1145, 1150, 1150, 1155, 1155, 1204, 1204, 1208, 1208, 1217, 1217, 1217, 1220, 1220, 1225, 1225, 1225, 1228, 1228, 1251, 1251, 1263, 1263, 1363, 1378, 1408, 1427, 1427, 1427, 1428, 1428, 1428, 1429, 1429, 1429, 1434, 1434, 1434, 1435, 1435, 1435, 1436, 1436, 1436, 1443, 1484, 1484, 1493, 1494, 1498, 1498, 1498, 1514, 1514 and 1583.
- 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 1452 and 1453.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 1345 and 1349.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 1346 and 1349.

CONCLUSION

We have audited the MASYA project released on October 2022 to discover issues and identify potential security vulnerabilities in MASYA 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 MASYA 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, weak sources of randomness, tx.origin as a part of authorization control 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. Werecommend avoiding"tx.origin" issue, the tx.origin environment variable has been found to influence a control flow decision. Note that using "tx.origin" as a security control might cause a situation where a user inadvertently authorizes a smart contract to perform an action on their behalf. It is recommended to use "msg.sender" instead and also don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

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.	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.	ISSUE FOUND
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 Oct 24 2022 03:28:03 GMT+0000 (Coordinated Universal Time)
Finished	Tuesday Oct 25 2022 01:47:46 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	MASYA.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	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-115	USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.	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-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 386

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
385   unchecked {
386   _approve(sender, _msgSender(), currentAllowance - amount);
387   }
388
389   return true;
390
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 405

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
404   function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
405   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
406   return true;
407   }
408
409
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 427

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
426   unchecked {
427   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
428   }
429
430   return true;
431
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 460

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
459   unchecked {
460   _balances[sender] = senderBalance - amount;
461   }
462   _balances[recipient] += amount;
463
464
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 462

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
461   }
462   _balances[recipient] += amount;
463
464   emit Transfer(sender, recipient, amount);
465
466
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 483

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
482
483   _totalSupply += amount;
484   _balances[account] += amount;
485   emit Transfer(address(0), account, amount);
486
487
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 484

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
483   _totalSupply += amount;
484   _balances[account] += amount;
485   emit Transfer(address(0), account, amount);
486
487   _afterTokenTransfer(address(0), account, amount);
488
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 509

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
508   unchecked {
509   _balances[account] = accountBalance - amount;
510   }
511   _totalSupply -= amount;
512
513
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 511

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
510   }
511   _totalSupply -= amount;
512
513   emit Transfer(account, address(0), amount);
514
515
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 607

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
606   unchecked {
607   uint256 c = a + b;
608   if (c < a) return (false, 0);
609   return (true, c);
610   }
611
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 621

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
620   if (b > a) return (false, 0);
621   return (true, a - b);
622   }
623   }
624
625
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 636

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
635   if (a == 0) return (true, 0);
636   uint256 c = a * b;
637   if (c / a != b) return (false, 0);
638   return (true, c);
639   }
640
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 637

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
636   uint256 c = a * b;
637   if (c / a != b) return (false, 0);
638   return (true, c);
639   }
640   }
641
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 650

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
649   if (b == 0) return (false, 0);
650   return (true, a / b);
651   }
652   }
653
654
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 662

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
661   if (b == 0) return (false, 0);
662   return (true, a % b);
663   }
664   }
665
666
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 677

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
676   function add(uint256 a, uint256 b) internal pure returns (uint256) {
677   return a + b;
678   }
679
680   /**
681
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 691

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
690   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
691   return a - b;
692   }
693
694   /**
695
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 705

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
704   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
705   return a * b;
706   }
707
708   /**
709
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 719

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
718   function div(uint256 a, uint256 b) internal pure returns (uint256) {
719   return a / b;
720   }
721
722   /**
723
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 735

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
734   function mod(uint256 a, uint256 b) internal pure returns (uint256) {
735   return a % b;
736   }
737
738   /**
739
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 758

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
757   require(b <= a, errorMessage);
758   return a - b;
759   }
760   }
761
762
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 781

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
780   require(b > 0, errorMessage);
781   return a / b;
782   }
783   }
784
785
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 807

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
806   require(b > 0, errorMessage);
807   return a % b;
808   }
809   }
810   }
811
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1141

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1140
1141   uint256 totalSupply = 1_000_000_000_000_000 * 1e18;
1142
1143   maxTransactionAmount = 20_000_000_000_000 * 1e18;
1144   maxWallet = 20_000_000_000_000 * 1e18;
1145
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1143

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1142
1143   maxTransactionAmount = 20_000_000_000_000 * 1e18;
1144   maxWallet = 20_000_000_000_000 * 1e18;
1145   swapTokensAtAmount = (totalSupply * 5) / 10000;
1146
1147
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1144

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1143   maxTransactionAmount = 20_000_000_000_000 * 1e18;
1144   maxWallet = 20_000_000_000_000 * 1e18;
1145   swapTokensAtAmount = (totalSupply * 5) / 10000;
1146
1147   buyMarketingFee = _buyMarketingFee;
1148
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1145

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1144   maxWallet = 20_000_000_000_000 * 1e18;
1145   swapTokensAtAmount = (totalSupply * 5) / 10000;
1146
1147   buyMarketingFee = _buyMarketingFee;
1148   buyLiquidityFee = _buyLiquidityFee;
1149
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1145

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1144   maxWallet = 20_000_000_000_000 * 1e18;
1145   swapTokensAtAmount = (totalSupply * 5) / 10000;
1146
1147   buyMarketingFee = _buyMarketingFee;
1148   buyLiquidityFee = _buyLiquidityFee;
1149
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1150

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1149   buyDevFee = _buyDevFee;
1150   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1151
1152   sellMarketingFee = _sellMarketingFee;
1153   sellLiquidityFee = _sellLiquidityFee;
1154
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1150

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1149   buyDevFee = _buyDevFee;
1150   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1151
1152   sellMarketingFee = _sellMarketingFee;
1153   sellLiquidityFee = _sellLiquidityFee;
1154
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1155

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1154   sellDevFee = _sellDevFee;
1155   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1156
1157   marketingWallet = address(0x7900d1Ce354015770822c245cEA7b4CCDCd73631); // set as
marketing wallet
1158   devWallet = address(0x0cefB0Fb76a7d2081c1905057aab2D21e07E4beB); // set as dev
wallet
1159
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1155

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1154   sellDevFee = _sellDevFee;
1155   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1156
1157   marketingWallet = address(0x7900d1Ce354015770822c245cEA7b4CCDCd73631); // set as
marketing wallet
1158   devWallet = address(0x0cefB0Fb76a7d2081c1905057aab2D21e07E4beB); // set as dev
wallet
1159
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1204

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1203   require(
1204   newAmount >= (totalSupply() * 1) / 100000,
1205   "Swap amount cannot be lower than 0.001% total supply."
1206   );
1207   require(
1208
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1204

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1203   require(
1204   newAmount >= (totalSupply() * 1) / 100000,
1205   "Swap amount cannot be lower than 0.001% total supply."
1206   );
1207   require(
1208
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1208

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1207   require(
1208   newAmount <= (totalSupply() * 5) / 1000,
1209   "Swap amount cannot be higher than 0.5% total supply."
1210   );
1211   swapTokensAtAmount = newAmount;
1212
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1208

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1207   require(
1208   newAmount <= (totalSupply() * 5) / 1000,
1209   "Swap amount cannot be higher than 0.5% total supply."
1210   );
1211   swapTokensAtAmount = newAmount;
1212
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1217

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1216   require(
1217   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1218   "Cannot set maxTransactionAmount lower than 0.1%"
1219   );
1220   maxTransactionAmount = newNum * (10**18);
1221
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1217

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1216   require(
1217   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1218   "Cannot set maxTransactionAmount lower than 0.1%"
1219   );
1220   maxTransactionAmount = newNum * (10**18);
1221
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1217

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1216   require(
1217   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1218   "Cannot set maxTransactionAmount lower than 0.1%"
1219   );
1220   maxTransactionAmount = newNum * (10**18);
1221
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1220

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1219   );
1220   maxTransactionAmount = newNum * (10**18);
1221   }
1222
1223   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1224
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1220

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1219   );
1220   maxTransactionAmount = newNum * (10**18);
1221   }
1222
1223   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1224
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1225

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1224   require(
1225   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1226   "Cannot set maxWallet lower than 0.5%"
1227   );
1228   maxWallet = newNum * (10**18);
1229
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1225

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1224   require(
1225   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1226   "Cannot set maxWallet lower than 0.5%"
1227   );
1228   maxWallet = newNum * (10**18);
1229
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1225

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1224   require(
1225   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1226   "Cannot set maxWallet lower than 0.5%"
1227   );
1228   maxWallet = newNum * (10**18);
1229
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1228

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1227   );
1228   maxWallet = newNum * (10**18);
1229   }
1230
1231   function excludeFromMaxTransaction(address updAds, bool isEx)
1232
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1228

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1227   );
1228   maxWallet = newNum * (10**18);
1229   }
1230
1231   function excludeFromMaxTransaction(address updAds, bool isEx)
1232
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1251

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1250   buyDevFee = _devFee;
1251   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1252   require(buyTotalFees <= 11, "Must keep fees at 11% or less");
1253   }
1254
1255
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1251

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1250   buyDevFee = _devFee;
1251   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1252   require(buyTotalFees <= 11, "Must keep fees at 11% or less");
1253   }
1254
1255
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1263

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1262   sellDevFee = _devFee;
1263   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1264   require(sellTotalFees <= 11, "Must keep fees at 11% or less");
1265   }
1266
1267
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1263

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1262   sellDevFee = _devFee;
1263   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1264   require(sellTotalFees <= 11, "Must keep fees at 11% or less");
1265   }
1266
1267
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1363

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1362   require(
1363   amount + balanceOf(to) <= maxWallet,
1364   "Max wallet exceeded"
1365   );
1366   }
1367
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1378

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1377   require(
1378   amount + balanceOf(to) <= maxWallet,
1379   "Max wallet exceeded"
1380   );
1381   }
1382
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1408

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1407   lpBurnEnabled &&
1408   block.timestamp >= lastLpBurnTime + lpBurnFrequency &&
1409   !_isExcludedFromFees[from]
1410   ) {
1411   autoBurnLiquidityPairTokens();
1412
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1427

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1426   fees = amount.mul(sellTotalFees).div(100);
1427   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1427

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1426   fees = amount.mul(sellTotalFees).div(100);
1427   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1427

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1426   fees = amount.mul(sellTotalFees).div(100);
1427   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1428

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1427   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431   // on buy
1432
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1428

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1427   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431   // on buy
1432
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1428

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1427   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431   // on buy
1432
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1429

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431   // on buy
1432   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1433
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1429

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431   // on buy
1432   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1433
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1429

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1428   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1429   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1430   }
1431   // on buy
1432   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1433
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1434

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1433   fees = amount.mul(buyTotalFees).div(100);
1434   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1434

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1433   fees = amount.mul(buyTotalFees).div(100);
1434   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1434

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1433   fees = amount.mul(buyTotalFees).div(100);
1434   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1435

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1434   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
1439
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1435

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1434   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
1439
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1435

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1434   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
1439
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1436

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
1439   if (fees > 0) {
1440
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1436

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
1439   if (fees > 0) {
1440
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1436

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1435   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1436   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1437   }
1438
1439   if (fees > 0) {
1440
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1443

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1442
1443   amount -= fees;
1444   }
1445
1446   super._transfer(from, to, amount);
1447
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1484

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1483   uint256 contractBalance = balanceOf(address(this));
1484   uint256 totalTokensToSwap = tokensForLiquidity +
1485   tokensForMarketing +
1486   tokensForDev;
1487   bool success;
1488
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1484

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1483   uint256 contractBalance = balanceOf(address(this));
1484   uint256 totalTokensToSwap = tokensForLiquidity +
1485   tokensForMarketing +
1486   tokensForDev;
1487   bool success;
1488
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1493

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1492
1493   if (contractBalance > swapTokensAtAmount * 20) {
1494   contractBalance = swapTokensAtAmount * 20;
1495   }
1496
1497
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1494

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1493   if (contractBalance > swapTokensAtAmount * 20) {
1494   contractBalance = swapTokensAtAmount * 20;
1495   }
1496
1497   // Halve the amount of liquidity tokens
1498
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1498

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1497   // Halve the amount of liquidity tokens
1498   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1499   totalTokensToSwap /
1500   2;
1501   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1502
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1498

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1497   // Halve the amount of liquidity tokens
1498   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1499   totalTokensToSwap /
1500   2;
1501   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1502
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1498

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1497   // Halve the amount of liquidity tokens
1498   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1499   totalTokensToSwap /
1500   2;
1501   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1502
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1514

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1513
1514   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1515
1516   tokensForLiquidity = 0;
1517   tokensForMarketing = 0;
1518
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1514

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1513
1514   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1515
1516   tokensForLiquidity = 0;
1517   tokensForMarketing = 0;
1518
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1583

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1582   require(
1583   block.timestamp > lastManualLpBurnTime + manualBurnFrequency,
1584   "Must wait for cooldown to finish"
1585   );
1586   require(percent <= 1000, "May not nuke more than 10% of tokens in LP");
1587
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 10

low SEVERITY

The current pragma Solidity directive is ""=0.8.10>=0.8.10>=0.8.0<0.9.0"". 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

- MASYA.sol

Locations

```
9   // SPDX-License-Identifier: MIT
10   pragma solidity =0.8.10 >=0.8.10 >=0.8.0 <0.9.0;
11   pragma experimental ABIEncoderV2;
12
13   ////// lib/openzeppelin-contracts/contracts/utils/Context.sol
14
```

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

LINE 1345

low SEVERITY

The tx.origin environment variable has been found to influence a control flow decision. Note that using "tx.origin" as a security control might cause a situation where a user inadvertently authorizes a smart contract to perform an action on their behalf. It is recommended to use "msg.sender" instead.

Source File

- MASYA.sol

Locations

```
1344   require(
1345   _holderLastTransferTimestamp[tx.origin] <
1346   block.number,
1347   "_transfer:: Transfer Delay enabled.  Only one purchase per block allowed."
1348   );
1349
```

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

LINE 1349

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

- MASYA.sol

Locations

```
1348   );
1349   _holderLastTransferTimestamp[tx.origin] = block.number;
1350   }
1351   }
1352
1353
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1452

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1451   address[] memory path = new address[](2);
1452   path[0] = address(this);
1453   path[1] = uniswapV2Router.WETH();
1454
1455   _approve(address(this), address(uniswapV2Router), tokenAmount);
1456
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1453

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1452   path[0] = address(this);
1453   path[1] = uniswapV2Router.WETH();
1454
1455   _approve(address(this), address(uniswapV2Router), tokenAmount);
1456
1457
```

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

LINE 1346

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1345   _holderLastTransferTimestamp[tx.origin] <
1346   block.number,
1347   "_transfer:: Transfer Delay enabled.  Only one purchase per block allowed."
1348   );
1349   _holderLastTransferTimestamp[tx.origin] = block.number;
1350
```

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

LINE 1349

low SEVERITY

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

Source File

- MASYA.sol

Locations

```
1348   );
1349   _holderLastTransferTimestamp[tx.origin] = block.number;
1350   }
1351   }
1352
1353
```

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.