Marvin Inu

Smart Contract Audit Report

SYSFIXED

21 Jan 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Marvin Inu	MARVIN	Binance Smart Chain

Addresses

Contract address	0x71ab195498b6dc1656abb4d9233f83ae5f19495b
Contract deployer address	0xA0b8ECa5Dc3af66A0dAA478d3006731e32258131

Project Website

https://marvin-ecosystem.com/

Codebase

https://bscscan.com/address/0x71ab195498b6dc1656abb4d9233f83ae5f19495b#code

SUMMARY

Marvin is here to stay, not only as a tribute to Elon's dog, but to bring user a full suite of treats, including his Launchpad, Staking, Farming, and more. Marvin INU has recognized a pattern of inadequate and unoriginal meme tokens, and aims to conquer them all!

Contract Summary

Documentation Quality

Marvin 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 Marvin Inu 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 481, 513, 536, 537, 572, 608, 674, 678, 690, 697, 706, 965, 965, 968, 969, 969, 970, 970, 975, 975, 980, 980, 1026, 1026, 1027, 1027, 1033, 1033, 1033, 1034, 1034, 1038, 1038, 1038, 1039, 1039, 1055, 1055, 1063, 1063, 1137, 1145, 1171, 1188, 1188, 1188, 1189, 1189, 1189, 1190, 1190, 1190, 1195, 1195, 1195, 1196, 1196, 1196, 1197, 1197, 1197, 1204, 1249, 1249, 1254, 1255, 1259, 1259, 1259, 1272, 1272 and 1321.
- 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 1214 and 1215.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 1129 and 1130.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 1129 and 1130.

CONCLUSION

We have audited the Marvin Inu project released on January 2023 to discover issues and identify potential security vulnerabilities in Marvin 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 satisfactory results with low-risk issues.

The Marvin Inu smart contract code issues do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some Use of "tx.origin" as a part of authorization control, Potential use of "block.number" as a source of randomness, and out-of-bounds array access in which the index access expression can cause an exception in case of the use of an invalid array index value. 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 act on their behalf. It is recommended to use "msg.sender" instead. 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 using 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.	PASS
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	Thursday Jan 20 2022 08:46:53 GMT+0000 (Coordinated Universal Time)
Finished	Friday Jan 21 2022 16:59:47 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	MarvinInu.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-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 481

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
480   function add(uint256 a, uint256 b) internal pure returns (uint256) {
481   uint256 c = a + b;
482   require(c >= a, "SafeMath: addition overflow");
483
484   return c;
485
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 513

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
512   require(b <= a, errorMessage);
513   uint256 c = a - b;
514
515   return c;
516   }
517
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 536

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
535
536   uint256 c = a * b;
537   require(c / a == b, "SafeMath: multiplication overflow");
538
539   return c;
540
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 537

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
536   uint256 c = a * b;
537   require(c / a == b, "SafeMath: multiplication overflow");
538
539   return c;
540   }
541
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 572

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
571   require(b > 0, errorMessage);
572   uint256 c = a / b;
573   // assert(a == b * c + a % b); // There is no case in which this doesn't hold
574
575   return c;
576
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 608

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
607   require(b != 0, errorMessage);
608   return a % b;
609   }
610   }
611
612
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 674

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
673   function mul(int256 a, int256 b) internal pure returns (int256) {
674   int256 c = a * b;
675
676   // Detect overflow when multiplying MIN_INT256 with -1
677   require(c != MIN_INT256 || (a & MIN_INT256) != (b & MIN_INT256));
678
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 678

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
677   require(c != MIN_INT256 || (a & MIN_INT256) != (b & MIN_INT256));
678   require((b == 0) || (c / b == a));
679   return c;
680   }
681
682
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 690

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
689   // Solidity already throws when dividing by 0.
690   return a / b;
691   }
692
693   /**
694
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 697

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
696   function sub(int256 a, int256 b) internal pure returns (int256) {
697   int256 c = a - b;
698   require((b >= 0 && c <= a) || (b < 0 && c > a));
699   return c;
700   }
701
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 706

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
705   function add(int256 a, int256 b) internal pure returns (int256) {
706   int256 c = a + b;
707   require((b >= 0 && c >= a) || (b < 0 && c < a));
708   return c;
709   }
710
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 965

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
964
965   uint256 totalSupply = 1 * 1e12 * 1e18;
966
967   //maxTransactionAmount = totalSupply * 50 / 1000; // 0.50% maxTransactionAmountTxn
968   maxTransactionAmount = 5000000000 * 1e18;
969
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 965

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
964
965   uint256 totalSupply = 1 * 1e12 * 1e18;
966
967   //maxTransactionAmount = totalSupply * 50 / 1000; // 0.50% maxTransactionAmountTxn
968   maxTransactionAmount = 5000000000 * 1e18;
969
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 968

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
967   //maxTransactionAmount = totalSupply * 50 / 1000; // 0.50% maxTransactionAmountTxn
968   maxTransactionAmount = 5000000000 * 1e18;
969   maxWallet = totalSupply * 15 / 1000; // 1.5% maxWallet
970   swapTokensAtAmount = totalSupply * 15 / 10000; // 0.15% swap wallet
971
972
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 969

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
968   maxTransactionAmount = 5000000000 * 1e18;
969   maxWallet = totalSupply * 15 / 1000; // 1.5% maxWallet
970   swapTokensAtAmount = totalSupply * 15 / 10000; // 0.15% swap wallet
971
972   buyMarketingFee = _buyMarketingFee;
973
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 969

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
968   maxTransactionAmount = 5000000000 * 1e18;
969   maxWallet = totalSupply * 15 / 1000; // 1.5% maxWallet
970   swapTokensAtAmount = totalSupply * 15 / 10000; // 0.15% swap wallet
971
972   buyMarketingFee = _buyMarketingFee;
973
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 970

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
969   maxWallet = totalSupply * 15 / 1000; // 1.5% maxWallet
970   swapTokensAtAmount = totalSupply * 15 / 10000; // 0.15% swap wallet
971
972   buyMarketingFee = _buyMarketingFee;
973   buyLiquidityFee = _buyLiquidityFee;
974
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 970

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
969   maxWallet = totalSupply * 15 / 1000; // 1.5% maxWallet
970   swapTokensAtAmount = totalSupply * 15 / 10000; // 0.15% swap wallet
971
972   buyMarketingFee = _buyMarketingFee;
973   buyLiquidityFee = _buyLiquidityFee;
974
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 975

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
974   buyDevFee = _buyDevFee;
975   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
976
977   sellMarketingFee = _sellMarketingFee;
978   sellLiquidityFee = _sellLiquidityFee;
979
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 975

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
974   buyDevFee = _buyDevFee;
975   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
976
977   sellMarketingFee = _sellMarketingFee;
978   sellLiquidityFee = _sellLiquidityFee;
979
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 980

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
979   sellDevFee = _sellDevFee;
980   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
981
982   marketingWallet = address(owner()); // set as marketing wallet
983   devWallet = address(owner()); // set as dev wallet
984
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 980

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
979   sellDevFee = _sellDevFee;
980   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
981
982   marketingWallet = address(owner()); // set as marketing wallet
983   devWallet = address(owner()); // set as dev wallet
984
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1026

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1025   function updateSwapTokensAtAmount(uint256 newAmount) external onlyOwner returns
(bool){
1026   require(newAmount >= totalSupply() * 1 / 100000, "Swap amount cannot be lower than
0.001% total supply.");
1027   require(newAmount <= totalSupply() * 5 / 1000, "Swap amount cannot be higher than
0.5% total supply.");
1028   swapTokensAtAmount = newAmount;
1029   return true;
1030
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1026

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1025   function updateSwapTokensAtAmount(uint256 newAmount) external onlyOwner returns
(bool){
1026   require(newAmount >= totalSupply() * 1 / 100000, "Swap amount cannot be lower than
0.001% total supply.");
1027   require(newAmount <= totalSupply() * 5 / 1000, "Swap amount cannot be higher than
0.5% total supply.");
1028   swapTokensAtAmount = newAmount;
1029   return true;
1030
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1027

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1026   require(newAmount >= totalSupply() * 1 / 100000, "Swap amount cannot be lower than
0.001% total supply.");
1027   require(newAmount <= totalSupply() * 5 / 1000, "Swap amount cannot be higher than
0.5% total supply.");
1028   swapTokensAtAmount = newAmount;
1029   return true;
1030   }
1031
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1027

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1026   require(newAmount >= totalSupply() * 1 / 100000, "Swap amount cannot be lower than
0.001% total supply.");
1027   require(newAmount <= totalSupply() * 5 / 1000, "Swap amount cannot be higher than
0.5% total supply.");
1028   swapTokensAtAmount = newAmount;
1029   return true;
1030   }
1031
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1033

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1032   function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
1033   require(newNum >= (totalSupply() * 1 / 1000)/1e18, "Cannot set
maxTransactionAmount lower than 0.1%");
1034   maxTransactionAmount = newNum * (10**18);
1035   }
1036
1037
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1033

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1032   function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
1033   require(newNum >= (totalSupply() * 1 / 1000)/1e18, "Cannot set
maxTransactionAmount lower than 0.1%");
1034   maxTransactionAmount = newNum * (10**18);
1035   }
1036
1037
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1033

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1032   function updateMaxTxnAmount(uint256 newNum) external onlyOwner {
1033   require(newNum >= (totalSupply() * 1 / 1000)/1e18, "Cannot set
maxTransactionAmount lower than 0.1%");
1034   maxTransactionAmount = newNum * (10**18);
1035   }
1036
1037
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1034

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1033   require(newNum >= (totalSupply() * 1 / 1000)/1e18, "Cannot set
maxTransactionAmount lower than 0.1%");
1034   maxTransactionAmount = newNum * (10**18);
1035   }
1036
1037   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1038
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1034

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1033   require(newNum >= (totalSupply() * 1 / 1000)/1e18, "Cannot set
maxTransactionAmount lower than 0.1%");
1034   maxTransactionAmount = newNum * (10**18);
1035   }
1036
1037   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1038
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1038

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1037   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1038   require(newNum >= (totalSupply() * 5 / 1000)/1e18, "Cannot set maxWallet lower
than 0.5%");
1039   maxWallet = newNum * (10**18);
1040   }
1041
1042
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1038

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1037   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1038   require(newNum >= (totalSupply() * 5 / 1000)/1e18, "Cannot set maxWallet lower
than 0.5%");
1039   maxWallet = newNum * (10**18);
1040   }
1041
1042
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1038

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1037   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1038   require(newNum >= (totalSupply() * 5 / 1000)/1e18, "Cannot set maxWallet lower
than 0.5%");
1039   maxWallet = newNum * (10**18);
1040   }
1041
1042
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1039

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1038   require(newNum >= (totalSupply() * 5 / 1000)/1e18, "Cannot set maxWallet lower
than 0.5%");
1039   maxWallet = newNum * (10**18);
1040   }
1041
1042   function excludeFromMaxTransaction(address updAds, bool isEx) public onlyOwner {
1043
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1039

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1038   require(newNum >= (totalSupply() * 5 / 1000)/1e18, "Cannot set maxWallet lower
than 0.5%");
1039   maxWallet = newNum * (10**18);
1040   }
1041
1042   function excludeFromMaxTransaction(address updAds, bool isEx) public onlyOwner {
1043
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1055

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1054   buyDevFee = _devFee;
1055   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1056   require(buyTotalFees <= 20, "Must keep fees at 20% or less");
1057   }
1058
1059
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1055

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1054   buyDevFee = _devFee;
1055   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1056   require(buyTotalFees <= 20, "Must keep fees at 20% or less");
1057   }
1058
1059
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1063

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1062   sellDevFee = _devFee;
1063   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1064   require(sellTotalFees <= 25, "Must keep fees at 25% or less");
1065   }
1066
1067
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1063

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1062   sellDevFee = _devFee;
1063   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1064   require(sellTotalFees <= 25, "Must keep fees at 25% or less");
1065   }
1066
1067
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1137

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1136   require(amount <= maxTransactionAmount, "Buy transfer amount exceeds the
maxTransactionAmount.");
1137   require(amount + balanceOf(to) <= maxWallet, "Max wallet exceeded");
1138   }
1139
1140   //when sell
1141
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1145

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1144   else if(!_isExcludedMaxTransactionAmount[to]){
1145   require(amount + balanceOf(to) <= maxWallet, "Max wallet exceeded");
1146   }
1147   }
1148   }
1149
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1171

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1170
1171   if(!swapping && automatedMarketMakerPairs[to] && lpBurnEnabled && block.timestamp
>= lastLpBurnTime + lpBurnFrequency && !_isExcludedFromFees[from]){
1172   autoBurnLiquidityPairTokens();
1173   }
1174
1175
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1188

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1187   fees = amount.mul(sellTotalFees).div(100);
1188   tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees;
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1188

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1187   fees = amount.mul(sellTotalFees).div(100);
1188   tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees;
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1188

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1187   fees = amount.mul(sellTotalFees).div(100);
1188   tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees;
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1189

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1188   tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees;
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192   // on buy
1193
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1189

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1188   tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees;
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192   // on buy
1193
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1189

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1188   tokensForLiquidity += fees * sellLiquidityFee / sellTotalFees;
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192   // on buy
1193
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1190

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192   // on buy
1193   else if(automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1194
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1190

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192   // on buy
1193   else if(automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1194
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1190

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1189   tokensForDev += fees * sellDevFee / sellTotalFees;
1190   tokensForMarketing += fees * sellMarketingFee / sellTotalFees;
1191   }
1192   // on buy
1193   else if(automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1194
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1195

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1194   fees = amount.mul(buyTotalFees).div(100);
1195   tokensForLiquidity += fees * buyLiquidityFee / buyTotalFees;
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1195

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1194   fees = amount.mul(buyTotalFees).div(100);
1195   tokensForLiquidity += fees * buyLiquidityFee / buyTotalFees;
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1195

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1194   fees = amount.mul(buyTotalFees).div(100);
1195   tokensForLiquidity += fees * buyLiquidityFee / buyTotalFees;
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1196

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1195   tokensForLiquidity += fees * buyLiquidityFee / buyTotalFees;
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
1200
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1196

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1195   tokensForLiquidity += fees * buyLiquidityFee / buyTotalFees;
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
1200
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1196

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1195   tokensForLiquidity += fees * buyLiquidityFee / buyTotalFees;
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
1200
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1197

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
1200   if(fees > 0){
1201
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1197

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
1200   if(fees > 0){
1201
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1197

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1196   tokensForDev += fees * buyDevFee / buyTotalFees;
1197   tokensForMarketing += fees * buyMarketingFee / buyTotalFees;
1198   }
1199
1200   if(fees > 0){
1201
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1204

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1203
1204   amount -= fees;
1205   }
1206
1207   super._transfer(from, to, amount);
1208
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1249

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1248   uint256 contractBalance = balanceOf(address(this));
1249   uint256 totalTokensToSwap = tokensForLiquidity + tokensForMarketing +
tokensForDev;
1250   bool success;
1251
1252   if(contractBalance == 0 || totalTokensToSwap == 0) {return;}
1253
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1249

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1248   uint256 contractBalance = balanceOf(address(this));
1249   uint256 totalTokensToSwap = tokensForLiquidity + tokensForMarketing +
tokensForDev;
1250   bool success;
1251
1252   if(contractBalance == 0 || totalTokensToSwap == 0) {return;}
1253
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1254

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1253
1254   if(contractBalance > swapTokensAtAmount * 20){
1255   contractBalance = swapTokensAtAmount * 20;
1256   }
1257
1258
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1255

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1254   if(contractBalance > swapTokensAtAmount * 20){
1255   contractBalance = swapTokensAtAmount * 20;
1256   }
1257
1258   // Halve the amount of liquidity tokens
1259
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1259

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1258   // Halve the amount of liquidity tokens
1259   uint256 liquidityTokens = contractBalance * tokensForLiquidity / totalTokensToSwap
/ 2;
1260   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1261
1262   uint256 initialETHBalance = address(this).balance;
1263
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1259

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1258   // Halve the amount of liquidity tokens
1259   uint256 liquidityTokens = contractBalance * tokensForLiquidity / totalTokensToSwap
/ 2;
1260   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1261
1262   uint256 initialETHBalance = address(this).balance;
1263
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1259

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1258   // Halve the amount of liquidity tokens
1259   uint256 liquidityTokens = contractBalance * tokensForLiquidity / totalTokensToSwap
/ 2;
1260   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1261
1262   uint256 initialETHBalance = address(this).balance;
1263
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1272

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1271
1272   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1273
1274
1275   tokensForLiquidity = 0;
1276
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1272

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1271
1272   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1273
1274
1275   tokensForLiquidity = 0;
1276
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1321

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1320   function manualBurnLiquidityPairTokens(uint256 percent) external onlyOwner returns
(bool){
1321   require(block.timestamp > lastManualLpBurnTime + manualBurnFrequency , "Must wait
for cooldown to finish");
1322   require(percent <= 1000, "May not nuke more than 10% of tokens in LP");
1323   lastManualLpBurnTime = block.timestamp;
1324
1325
```

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

LINE 1129

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

- MarvinInu.sol

Locations

```
1128   if (to != owner() && to != address(uniswapV2Router) && to !=
address(uniswapV2Pair)){
1129   require(_holderLastTransferTimestamp[tx.origin] < block.number, "_transfer::
Transfer Delay enabled.  Only one purchase per block allowed.");
1130   _holderLastTransferTimestamp[tx.origin] = block.number;
1131   }
1132   }
1133
```

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

LINE 1130

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

- MarvinInu.sol

Locations

```
1129   require(_holderLastTransferTimestamp[tx.origin] < block.number, "_transfer::
Transfer Delay enabled.  Only one purchase per block allowed.");
1130   _holderLastTransferTimestamp[tx.origin] = block.number;
1131   }
1132   }
1133
1134
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1214

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1213   address[] memory path = new address[](2);
1214   path[0] = address(this);
1215   path[1] = uniswapV2Router.WETH();
1216
1217   _approve(address(this), address(uniswapV2Router), tokenAmount);
1218
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1215

low SEVERITY

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

Source File

- MarvinInu.sol

Locations

```
1214   path[0] = address(this);
1215   path[1] = uniswapV2Router.WETH();
1216
1217   _approve(address(this), address(uniswapV2Router), tokenAmount);
1218
1219
```

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

LINE 1129

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

- MarvinInu.sol

Locations

```
1128   if (to != owner() && to != address(uniswapV2Router) && to !=
address(uniswapV2Pair)){
1129   require(_holderLastTransferTimestamp[tx.origin] < block.number, "_transfer::
Transfer Delay enabled.  Only one purchase per block allowed.");
1130   _holderLastTransferTimestamp[tx.origin] = block.number;
1131   }
1132   }
1133
```

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

LINE 1130

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

- MarvinInu.sol

Locations

```
1129   require(_holderLastTransferTimestamp[tx.origin] < block.number, "_transfer::
Transfer Delay enabled.  Only one purchase per block allowed.");
1130   _holderLastTransferTimestamp[tx.origin] = block.number;
1131   }
1132   }
1133
1134
```

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.