Baby Moon Floki

Smart Contract Audit Report

SYSFIXED

04 Oct 2021

TABLE OF CONTENTS

Audited Details

- Audited Project
- Blockchain
- Addresses
- Project Website
- Codebase

Summary

- Contract Summary
- Audit Findings Summary
- Vulnerabilities Summary

Conclusion

Audit Results

Smart Contract Analysis

- Detected Vulnerabilities

Disclaimer

About Us

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Baby Moon Floki	Floki	Binance Smart Chain

Addresses

Contract address	0x54e87ed5a096f09d9665fd114002bddfc2084a7f
Contract deployer address	0x75001CCDa5B6a711546D9BC14Ac805Dd78Ccc24f

Project Website

https://www.decubate.com/

Codebase

https://bscscan.com/address/0x54e87ed5a096f09d9665fd114002bddfc2084a7f#contracts

SUMMARY

Baby Moon Floki is the gem for meme coin investors looking for hype around new puppy Elon. Called by his beloved father, Elon Musk, the potential of this adorable little puppy is limitless. Our lovely baby was finally given the name given to him by Elon Musk himself, and this is "Floki". Floki has always dreamed of becoming the first dog to put his paws on the moon. The Baby Moon Floki community intends to bring it there! Join us!

Contract Summary

Documentation Quality

Baby Moon Floki 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 Baby Moon Floki with the discovery of several low issues.

Test Coverage

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

Audit Findings Summary

- SWC-100 SWC-108 | Explicitly define visibility for all state variables on lines 493.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 42, 54, 64, 65, 77, 89, 196, 434, 434, 434, 434, 435, 435, 470, 479, 479, 479, 479, 480, 480, 480, 480, 481, 481, 488, 655, 657, 700, 715, 718, 722, 735, 735, 735, 735, 955, 974, 980, 1025, 1027, 1034, 1038, 1040, 1048, 1060, 1073, 1133, 1133, 1133, 1133, 1140, 1140, 1140, 1140 and 657.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 11.
- 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 484, 656, 657, 657, 720, 721, 814, 815, 834, 835, 956, 956, 957, 958, 1015, 1016, 1020, 1029, 1031, 1031, 1032 and 1032.

CONCLUSION

We have audited the Baby Moon Floki project released on October 2024 to discover issues and identify potential security vulnerabilities in Baby Moon Floki Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

The security audit report provides satisfactory results with low-risk issues.

The issues found in the Baby Moon Floki 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, state variable visibility is not set, public state variable with array type causing reachable exception by default, and out-of-bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value. The current pragma Solidity directive is ""^0.8.4"". Specifying a fixed compiler version is recommended to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code. It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwapAndLiquify" is internal. Other possible visibility settings are public and private.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	ISSUE FOUND
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	ISSUE FOUND
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	PASS
Unprotected Ether Withdrawal	SWC-105	Due to missing or insufficient access controls, malicious parties can withdraw from the contract.	PASS
SELFDESTRUCT Instruction	SWC-106	The contract should not be self-destructible while it has funds belonging to users.	PASS
Reentrancy	SWC-107	Check effect interaction pattern should be followed if the code performs recursive call.	PASS
Uninitialized Storage Pointer	SWC-109	Uninitialized local storage variables can point to unexpected storage locations in the contract.	PASS
Assert Violation	SWC-110 SWC-123	Properly functioning code should never reach a failing assert statement.	ISSUE FOUND
Deprecated Solidity Functions	SWC-111	Deprecated built-in functions should never be used.	PASS
Delegate call to Untrusted Callee	SWC-112	Delegatecalls should only be allowed to trusted addresses.	PASS

DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS
Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	PASS
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Incorrect Constructor Name	SWC-118	Constructors are special functions that are called only once during the contract creation.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	PASS
Write to Arbitrary Storage Location	SWC-124	The contract is responsible for ensuring that only authorized user or contract accounts may write to sensitive storage locations.	PASS
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they have identical functions, a developer should carefully specify inheritance in the correct order. The rule of thumb is to inherit contracts from more /general/ to more /specific/.	PASS
Insufficient Gas Griefing	SWC-126	Insufficient gas griefing attacks can be performed on contracts which accept data and use it in a sub-call on another contract.	PASS
Arbitrary Jump Function	SWC-127	As Solidity doesnt support pointer arithmetics, it is impossible to change such variable to an arbitrary value.	PASS

Typographical Error	SWC-129	A typographical error can occur for example when the intent of a defined operation is to sum a number to a variable.	PASS
Override control character	SWC-130	Malicious actors can use the Right-To-Left-Override unicode character to force RTL text rendering and confuse users as to the real intent of a contract.	PASS
Unused variables	SWC-131 SWC-135	Unused variables are allowed in Solidity and they do not pose a direct security issue.	PASS
Unexpected Ether balance	SWC-132	Contracts can behave erroneously when they strictly assume a specific Ether balance.	PASS
Hash Collisions Variable	SWC-133	Using abi.encodePacked() with multiple variable length arguments can, in certain situations, lead to a hash collision.	PASS
Hardcoded gas amount	SWC-134	The transfer() and send() functions forward a fixed amount of 2300 gas.	PASS
Unencrypted Private Data	SWC-136	It is a common misconception that private type variables cannot be read.	PASS

SMART CONTRACT ANALYSIS

Started	Sunday Oct 03 2021 01:27:58 GMT+0000 (Coordinated Universal Time)
Finished	Monday Oct 04 2021 06:39:25 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	BabyMoonFloki.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	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-110	PUBLIC STATE VARIABLE WITH ARRAY TYPE CAUSING REACHABLE EXCEPTION BY DEFAULT.	low	acknowledged

SYSFIXED

SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 42

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
41   function add(uint256 a, uint256 b) internal pure returns (uint256) {
42   uint256 c = a + b;
43   require(c >= a, "SafeMath: addition overflow");
44
45   return c;
46
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 54

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
53   require(b <= a, errorMessage);
54   uint256 c = a - b;
55
56   return c;
57   }
58
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 64

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
63
64   uint256 c = a * b;
65   require(c / a == b, "SafeMath: multiplication overflow");
66
67   return c;
68
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 65

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
64   uint256 c = a * b;
65   require(c / a == b, "SafeMath: multiplication overflow");
66
67   return c;
68   }
69
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 77

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
76   require(b > 0, errorMessage);
77   uint256 c = a / b;
78   // assert(a == b * c + a % b); // There is no case in which this doesn't hold
79
80   return c;
81
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 89

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
88   require(b != 0, errorMessage);
89   return a % b;
90   }
91   }
92
93
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 196

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
195   _owner = address(0);
196   _lockTime = block.timestamp + time;
197   emit OwnershipTransferred(_owner, address(0));
198   }
199
200
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
433   uint256 private constant MAX = ~uint256(0);
434   uint256 private _tTotal = 100000000000 * 10**6 * 10**9;
435   uint256 private _rTotal = (MAX - (MAX % _tTotal));
436   uint256 private _tFeeTotal;
437
438
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
433   uint256 private constant MAX = ~uint256(0);
434   uint256 private _tTotal = 100000000000 * 10**6 * 10**9;
435   uint256 private _rTotal = (MAX - (MAX % _tTotal));
436   uint256 private _tFeeTotal;
437
438
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
433   uint256 private constant MAX = ~uint256(0);
434   uint256 private _tTotal = 100000000000 * 10**6 * 10**9;
435   uint256 private _rTotal = (MAX - (MAX % _tTotal));
436   uint256 private _tFeeTotal;
437
438
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 434

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
433   uint256 private constant MAX = ~uint256(0);
434   uint256 private _tTotal = 100000000000 * 10**6 * 10**9;
435   uint256 private _rTotal = (MAX - (MAX % _tTotal));
436   uint256 private _tFeeTotal;
437
438
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 435

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
434   uint256 private _tTotal = 100000000000 * 10**6 * 10**9;
435   uint256 private _rTotal = (MAX - (MAX % _tTotal));
436   uint256 private _tFeeTotal;
437
438   string private _name = "Baby Moon Floki";
439
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 435

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
434   uint256 private _tTotal = 100000000000 * 10**6 * 10**9;
435   uint256 private _rTotal = (MAX - (MAX % _tTotal));
436   uint256 private _tFeeTotal;
437
438   string private _name = "Baby Moon Floki";
439
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 470

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
469   uint256 public _startTimeForSwap;
470   uint256 public _intervalMinutesForSwap = 1 * 1 minutes;
471
472   uint256 public _buyBackRangeRate = 80;
473
474
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
478
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
478
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
478
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
478
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 480

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483   // LookBack into historical sale data
484
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 480

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483   // LookBack into historical sale data
484
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 480

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483   // LookBack into historical sale data
484
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 480

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
479   uint256 public _maxTxAmount = 600000000 * 10**6 * 10**9;
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483   // LookBack into historical sale data
484
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 481

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483   // LookBack into historical sale data
484   SellHistories[] public _sellHistories;
485
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 481

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
480   uint256 private minimumTokensBeforeSwap = 40000000 * 10**6 * 10**9;
481   uint256 public buyBackSellLimit = 1 * 10**13;
482
483   // LookBack into historical sale data
484   SellHistories[] public _sellHistories;
485
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 488

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
487   uint256 public _buyBackTimeInterval = 5 minutes;
488   uint256 public _buyBackMaxTimeForHistories = 24 * 60 minutes;
489
490   IUniswapV2Router02 public uniswapV2Router;
491   address public uniswapV2Pair;
492
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 655

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
654   require(_isExcluded[account], "Account is not excluded");
655   for (uint256 i = 0; i < _excluded.length; i++) {
656   if (_excluded[i] == account) {
657   _excluded[i] = _excluded[_excluded.length - 1];
658   _tOwned[account] = 0;
659
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 657

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
656   if (_excluded[i] == account) {
657   _excluded[i] = _excluded[_excluded.length - 1];
658   _tOwned[account] = 0;
659   _isExcluded[account] = false;
660   _excluded.pop();
661
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 700

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
699   if (to == uniswapV2Pair) {
700   if (overMinimumTokenBalance && _startTimeForSwap + _intervalMinutesForSwap <=
block.timestamp) {
701   _startTimeForSwap = block.timestamp;
702   contractTokenBalance = minimumTokensBeforeSwap;
703   swapTokens(contractTokenBalance);
704
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 715

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
714   uint256 sumBnbAmount = 0;
715   uint256 startTime = block.timestamp - _buyBackTimeInterval;
716   uint256 cnt = 0;
717
718   for (uint i = 0; i < _sellHistories.length; i ++) {
719
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 718

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
717
718   for (uint i = 0; i < _sellHistories.length; i ++) {
719
720   if (_sellHistories[i].time >= startTime) {
721   sumBnbAmount = sumBnbAmount.add(_sellHistories[i].bnbAmount);
722
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 722

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
721   sumBnbAmount = sumBnbAmount.add(_sellHistories[i].bnbAmount);
722   cnt = cnt + 1;
723   }
724   }
725
726
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 735

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
734
735   uint256 _bBSLimit = _bBSLimitMin +
uint256(keccak256(abi.encodePacked(block.timestamp, block.difficulty))) % (_bBSLimitMax -
_bBSLimitMin + 1);
736
737   if (balance > _bBSLimit) {
738   buyBackTokens(_bBSLimit);
739
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 735

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
734
735   uint256 _bBSLimit = _bBSLimitMin +
uint256(keccak256(abi.encodePacked(block.timestamp, block.difficulty))) % (_bBSLimitMax -
_bBSLimitMin + 1);
736
737   if (balance > _bBSLimit) {
738   buyBackTokens(_bBSLimit);
739
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 735

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
734
735   uint256 _bBSLimit = _bBSLimitMin +
uint256(keccak256(abi.encodePacked(block.timestamp, block.difficulty))) % (_bBSLimitMax -
_bBSLimitMin + 1);
736
737   if (balance > _bBSLimit) {
738   buyBackTokens(_bBSLimit);
739
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 735

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
734
735   uint256 _bBSLimit = _bBSLimitMin +
uint256(keccak256(abi.encodePacked(block.timestamp, block.difficulty))) % (_bBSLimitMax -
_bBSLimitMin + 1);
736
737   if (balance > _bBSLimit) {
738   buyBackTokens(_bBSLimit);
739
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 955

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
954   uint256 tSupply = _tTotal;
955   for (uint256 i = 0; i < _excluded.length; i++) {
956   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
957   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
958   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
959
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 974

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
973   return _amount.mul(_taxFee).div(
974   10**2
975   );
976   }
977
978
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 980

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
979   return _amount.mul(_liquidityFee).div(
980   10**2
981   );
982   }
983
984
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1025

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1024   uint256 i = 0;
1025   uint256 maxStartTimeForHistories = block.timestamp - _buyBackMaxTimeForHistories;
1026
1027   for (uint256 j = 0; j < _sellHistories.length; j ++) {
1028
1029
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1027

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1026
1027   for (uint256 j = 0; j < _sellHistories.length; j ++) {
1028
1029   if (_sellHistories[j].time >= maxStartTimeForHistories) {
1030
1031
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1034

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1033
1034   i = i + 1;
1035   }
1036   }
1037
1038
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1038

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1037
1038   uint256 removedCnt = _sellHistories.length - i;
1039
1040   for (uint256 j = 0; j < removedCnt; j ++) {
1041
1042
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1040

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1039
1040   for (uint256 j = 0; j < removedCnt; j ++) {
1041
1042   _sellHistories.pop();
1043   }
1044
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1048

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1047   function SetBuyBackMaxTimeForHistories(uint256 newMinutes) external onlyOwner {
1048   _buyBackMaxTimeForHistories = newMinutes * 1 minutes;
1049   }
1050
1051   function SetBuyBackDivisor(uint256 newDivisor) external onlyOwner {
1052
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1060

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1059   function SetBuyBackTimeInterval(uint256 newMinutes) external onlyOwner {
1060   _buyBackTimeInterval = newMinutes * 1 minutes;
1061   }
1062
1063   function SetBuyBackRangeRate(uint256 newPercent) external onlyOwner {
1064
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1073

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1072   function SetSwapMinutes(uint256 newMinutes) external onlyOwner {
1073   _intervalMinutesForSwap = newMinutes * 1 minutes;
1074   }
1075
1076   function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
1077
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1133

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1132   _liquidityFee = 0;
1133   _maxTxAmount = 1000000000 * 10**6 * 10**9;
1134   }
1135
1136   function afterPreSale() external onlyOwner {
1137
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1133

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1132   _liquidityFee = 0;
1133   _maxTxAmount = 1000000000 * 10**6 * 10**9;
1134   }
1135
1136   function afterPreSale() external onlyOwner {
1137
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1133

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1132   _liquidityFee = 0;
1133   _maxTxAmount = 1000000000 * 10**6 * 10**9;
1134   }
1135
1136   function afterPreSale() external onlyOwner {
1137
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1133

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1132   _liquidityFee = 0;
1133   _maxTxAmount = 1000000000 * 10**6 * 10**9;
1134   }
1135
1136   function afterPreSale() external onlyOwner {
1137
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1140

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1139   _liquidityFee = 10;
1140   _maxTxAmount = 3000000 * 10**6 * 10**9;
1141   }
1142
1143   function transferToAddressETH(address payable recipient, uint256 amount) private {
1144
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1140

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1139   _liquidityFee = 10;
1140   _maxTxAmount = 3000000 * 10**6 * 10**9;
1141   }
1142
1143   function transferToAddressETH(address payable recipient, uint256 amount) private {
1144
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1140

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1139   _liquidityFee = 10;
1140   _maxTxAmount = 3000000 * 10**6 * 10**9;
1141   }
1142
1143   function transferToAddressETH(address payable recipient, uint256 amount) private {
1144
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1140

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1139   _liquidityFee = 10;
1140   _maxTxAmount = 3000000 * 10**6 * 10**9;
1141   }
1142
1143   function transferToAddressETH(address payable recipient, uint256 amount) private {
1144
```

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

LINE 657

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
656   if (_excluded[i] == account) {
657   _excluded[i] = _excluded[_excluded.length - 1];
658   _tOwned[account] = 0;
659   _isExcluded[account] = false;
660   _excluded.pop();
661
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 11

low SEVERITY

The current pragma Solidity directive is ""^0.8.4"". 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

- BabyMoonFloki.sol

Locations

```
10
11   pragma solidity ^0.8.4;
12
13   abstract contract Context {
14   function _msgSender() internal view virtual returns (address payable) {
15
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 493

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
492
493   bool inSwapAndLiquify;
494   bool public swapAndLiquifyEnabled = false;
495   bool public buyBackEnabled = true;
496
497
```

SWC-110 | PUBLIC STATE VARIABLE WITH ARRAY TYPE CAUSING REACHABLE EXCEPTION BY DEFAULT.

LINE 484

low SEVERITY

The public state variable "_sellHistories" in "BabyMoonFloki" contract has type "struct BabyMoonFloki.SellHistories[]" and can cause an exception in case of use of invalid array index value.

Source File

- BabyMoonFloki.sol

Locations

```
483   // LookBack into historical sale data
484   SellHistories[] public _sellHistories;
485   bool public _isAutoBuyBack = true;
486   uint256 public _buyBackDivisor = 10;
487   uint256 public _buyBackTimeInterval = 5 minutes;
488
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 656

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
655   for (uint256 i = 0; i < _excluded.length; i++) {
656   if (_excluded[i] == account) {
657   _excluded[i] = _excluded[_excluded.length - 1];
658   _tOwned[account] = 0;
659   _isExcluded[account] = false;
660
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 657

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
656   if (_excluded[i] == account) {
657   _excluded[i] = _excluded[_excluded.length - 1];
658   _tOwned[account] = 0;
659   _isExcluded[account] = false;
660   _excluded.pop();
661
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 657

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
656   if (_excluded[i] == account) {
657   _excluded[i] = _excluded[_excluded.length - 1];
658   _tOwned[account] = 0;
659   _isExcluded[account] = false;
660   _excluded.pop();
661
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 720

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
719
720   if (_sellHistories[i].time >= startTime) {
721   sumBnbAmount = sumBnbAmount.add(_sellHistories[i].bnbAmount);
722   cnt = cnt + 1;
723   }
724
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 721

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
720   if (_sellHistories[i].time >= startTime) {
721   sumBnbAmount = sumBnbAmount.add(_sellHistories[i].bnbAmount);
722   cnt = cnt + 1;
723   }
724   }
725
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 814

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
813   address[] memory path = new address[](2);
814   path[0] = address(this);
815   path[1] = uniswapV2Router.WETH();
816
817   _approve(address(this), address(uniswapV2Router), tokenAmount);
818
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 815

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
814   path[0] = address(this);
815   path[1] = uniswapV2Router.WETH();
816
817   _approve(address(this), address(uniswapV2Router), tokenAmount);
818
819
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 834

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
833   address[] memory path = new address[](2);
834   path[0] = uniswapV2Router.WETH();
835   path[1] = address(this);
836
837   // Make the swap
838
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 835

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
834   path[0] = uniswapV2Router.WETH();
835   path[1] = address(this);
836
837   // Make the swap
838   uniswapV2Router.swapExactETHForTokensSupportingFeeOnTransferTokens{value: amount}(
839
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 956

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
955   for (uint256 i = 0; i < _excluded.length; i++) {
956   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
957   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
958   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
959   }
960
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 956

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
955   for (uint256 i = 0; i < _excluded.length; i++) {
956   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
957   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
958   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
959   }
960
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 957

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
956   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
957   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
958   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
959   }
960   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
961
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 958

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
957   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
958   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
959   }
960   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
961   return (rSupply, tSupply);
962
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1015

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1014
1015   path[0] = address(this);
1016   path[1] = uniswapV2Router.WETH();
1017
1018   uint[] memory amounts = uniswapV2Router.getAmountsOut(tokenAmount, path);
1019
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1016

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1015   path[0] = address(this);
1016   path[1] = uniswapV2Router.WETH();
1017
1018   uint[] memory amounts = uniswapV2Router.getAmountsOut(tokenAmount, path);
1019
1020
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1020

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1019
1020   return amounts[1];
1021   }
1022
1023   function _removeOldSellHistories() private {
1024
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1029

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1028
1029   if (_sellHistories[j].time >= maxStartTimeForHistories) {
1030
1031   _sellHistories[i].time = _sellHistories[j].time;
1032   _sellHistories[i].bnbAmount = _sellHistories[j].bnbAmount;
1033
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1031

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1030
1031   _sellHistories[i].time = _sellHistories[j].time;
1032   _sellHistories[i].bnbAmount = _sellHistories[j].bnbAmount;
1033
1034   i = i + 1;
1035
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1031

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1030
1031   _sellHistories[i].time = _sellHistories[j].time;
1032   _sellHistories[i].bnbAmount = _sellHistories[j].bnbAmount;
1033
1034   i = i + 1;
1035
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1032

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1031   _sellHistories[i].time = _sellHistories[j].time;
1032   _sellHistories[i].bnbAmount = _sellHistories[j].bnbAmount;
1033
1034   i = i + 1;
1035   }
1036
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1032

low SEVERITY

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

Source File

- BabyMoonFloki.sol

Locations

```
1031   _sellHistories[i].time = _sellHistories[j].time;
1032   _sellHistories[i].bnbAmount = _sellHistories[j].bnbAmount;
1033
1034   i = i + 1;
1035   }
1036
```

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.