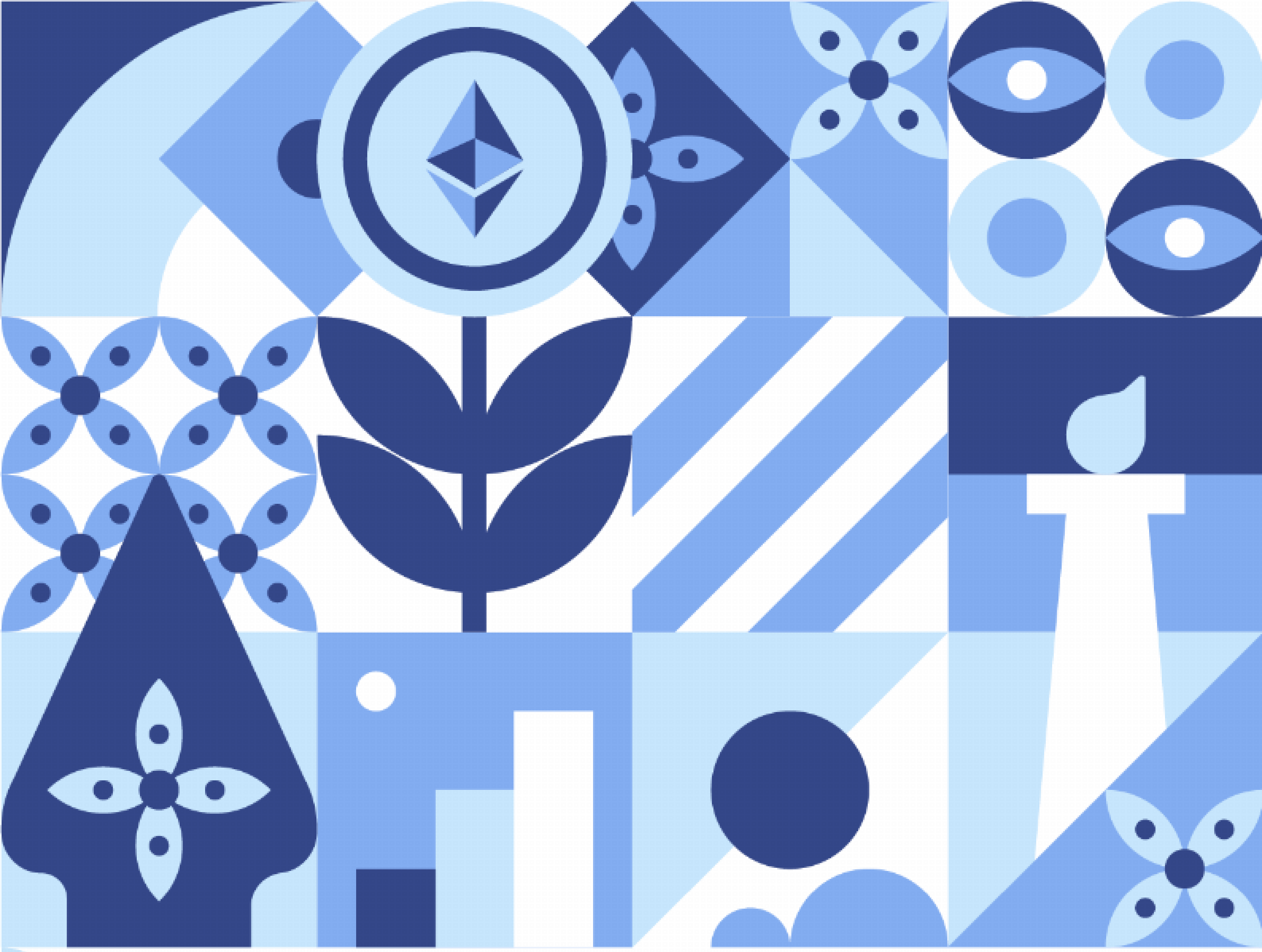

Archie Neko

Smart Contract Audit Report

SYSFIXED

25 Mar 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Archie Neko	ARCHIE	Ethereum

Addresses

Contract address	0xFE5F69dfa2d4501E78078266F6d430c079098f90
Contract deployer address	0x088B1fb7919Fe8E25377590AC759d4cD73378400

Project Website

https://www.archieneko.com/

Codebase

https://etherscan.io/address/0xFE5F69dfa2d4501E78078266F6d430c079098f90#code

SUMMARY

The vision of Archie Neko is to generate wealth for a decentralized community in a safe and secure environment, leveraging DeFi protocols and to impact the globe for good, thereby cherishing the memory of our beloved Archie.

Contract Summary

Documentation Quality

Archie Neko 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 Archie Neko 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 429 and 430.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 35, 46, 56, 57, 67, 78, 438, 438, 438, 438, 439, 439, 475, 476, 478, 478, 479, 479, 634, 640, 646, 646, 657, 664, 666, 670, 671, 672, 709, 709, 715, 715, 715, 716, 716, 717, 717, 718, 718, 719, 719, 723, 723, 723, 727, 727, 728, 728, 729, 729, 730, 730, 894, 903, 912, 921, 961, 972, 978, 1060, 1062, 1144, 1144, 1145, 1145, 1159, 1176, 1176, 1187, 1187, 1188 and 1062.
- 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 749, 750, 962, 962, 963, 964, 1061, 1062, 1062, 1187 and 1187.

CONCLUSION

We have audited the Archie Neko project released on March 2022 to discover issues and identify potential security vulnerabilities in Archie Neko Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

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

The issues found in the Archie Neko 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 state variable visibility is not set and out of bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	ISSUE FOUND
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	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.	PASS
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Incorrect Constructor Name	SWC-118	Constructors are special functions that are called only once during the contract creation.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	PASS
Write to Arbitrary Storage Location	SWC-124	The contract is responsible for ensuring that only authorized user or contract accounts may write to sensitive storage locations.	PASS
Incorrect Inheritance Order	SWC-125	When inheriting multiple contracts, especially if they have identical functions, a developer should carefully specify inheritance in the correct order. The rule of thumb is to inherit contracts from more /general/ to more /specific/.	PASS
Insufficient Gas Griefing	SWC-126	Insufficient gas griefing attacks can be performed on contracts which accept data and use it in a sub-call on another contract.	PASS
Arbitrary Jump Function	SWC-127	As Solidity doesnt support pointer arithmetics, it is impossible to change such variable to an arbitrary value.	PASS

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

SMART CONTRACT ANALYSIS

Started	Thursday Mar 24 2022 04:46:19 GMT+0000 (Coordinated Universal Time)
Finished	Friday Mar 25 2022 16:35:35 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	ArhcieNeko.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	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 35

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
34   function add(uint256 a, uint256 b) internal pure returns (uint256) {
35   uint256 c = a + b;
36   require(c >= a, "SafeMath: addition overflow");
37   return c;
38   }
39
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 46

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
45   require(b <= a, errorMessage);
46   uint256 c = a - b;
47   return c;
48   }
49
50
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 56

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
55
56   uint256 c = a * b;
57   require(c / a == b, "SafeMath: multiplication overflow");
58   return c;
59   }
60
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 57

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
56   uint256 c = a * b;
57   require(c / a == b, "SafeMath: multiplication overflow");
58   return c;
59   }
60
61
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 67

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
66   require(b > 0, errorMessage);
67   uint256 c = a / b;
68
69   return c;
70   }
71
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 78

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
77   require(b != 0, errorMessage);
78   return a % b;
79   }
80   }
81
82
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
437   uint256 private constant MAX = ~uint256(0);
438   uint256 private _tTotal = 10 * 10 ** 21 * 10 ** _decimals;//
439   uint256 private _rTotal = (MAX - (MAX % _tTotal));
440   uint256 private _tFeeTotal;
441
442
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
437   uint256 private constant MAX = ~uint256(0);
438   uint256 private _tTotal = 10 * 10 ** 21 * 10 ** _decimals;//
439   uint256 private _rTotal = (MAX - (MAX % _tTotal));
440   uint256 private _tFeeTotal;
441
442
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
437   uint256 private constant MAX = ~uint256(0);
438   uint256 private _tTotal = 10 * 10 ** 21 * 10 ** _decimals;//
439   uint256 private _rTotal = (MAX - (MAX % _tTotal));
440   uint256 private _tFeeTotal;
441
442
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 438

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
437   uint256 private constant MAX = ~uint256(0);
438   uint256 private _tTotal = 10 * 10 ** 21 * 10 ** _decimals;//
439   uint256 private _rTotal = (MAX - (MAX % _tTotal));
440   uint256 private _tFeeTotal;
441
442
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 439

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
438   uint256 private _tTotal = 10 * 10 ** 21 * 10 ** _decimals;//
439   uint256 private _rTotal = (MAX - (MAX % _tTotal));
440   uint256 private _tFeeTotal;
441
442   uint256 public _liquidityTax;
443
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 439

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
438   uint256 private _tTotal = 10 * 10 ** 21 * 10 ** _decimals;//
439   uint256 private _rTotal = (MAX - (MAX % _tTotal));
440   uint256 private _tFeeTotal;
441
442   uint256 public _liquidityTax;
443
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 475

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
474
475   uint256 private numTokensSellToAddToLiquidity = _tTotal / 500;
476   uint256 public _maxwalletamount = _tTotal / 100;
477   uint256 public _dailyTimeLimit = 24 hours;
478   uint256 public _dailymaxTxAmount = 24 * 10 ** 18;
479
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 476

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
475   uint256 private numTokensSellToAddToLiquidity = _tTotal / 500;
476   uint256 public _maxwalletamount = _tTotal / 100;
477   uint256 public _dailyTimeLimit = 24 hours;
478   uint256 public _dailymaxTxAmount = 24 * 10 ** 18;
479   uint256 public _maxTxAmount = 3 * 10 ** 18;
480
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 478

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
477   uint256 public _dailyTimeLimit = 24 hours;
478   uint256 public _dailymaxTxAmount = 24 * 10 ** 18;
479   uint256 public _maxTxAmount = 3 * 10 ** 18;
480
481   uint256 public _tokenAmountForMarketingTax;
482
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 478

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
477   uint256 public _dailyTimeLimit = 24 hours;
478   uint256 public _dailymaxTxAmount = 24 * 10 ** 18;
479   uint256 public _maxTxAmount = 3 * 10 ** 18;
480
481   uint256 public _tokenAmountForMarketingTax;
482
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
478   uint256 public _dailymaxTxAmount = 24 * 10 ** 18;
479   uint256 public _maxTxAmount = 3 * 10 ** 18;
480
481   uint256 public _tokenAmountForMarketingTax;
482   uint256 public _tokenAmountForFoundationTax;
483
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 479

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
478   uint256 public _dailymaxTxAmount = 24 * 10 ** 18;
479   uint256 public _maxTxAmount = 3 * 10 ** 18;
480
481   uint256 public _tokenAmountForMarketingTax;
482   uint256 public _tokenAmountForFoundationTax;
483
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 634

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
633   require(amount > 0, "Transfer amount must be greater than zero");
634   if(transactionData[from].isLocked && block.timestamp >=
transactionData[from].lockedTime + transactionData[from].lockPeriod) {
635   transactionData[from].isLocked = false;
636   transactionData[from].lockPeriod = 0;
637   transactionData[from].lockedTime = 0;
638
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 640

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
639
640   if (transactionData[from].isLocked && block.timestamp <
transactionData[from].lockedTime + transactionData[from].lockPeriod) {
641   require(!transactionData[from].isLocked, "Locked Account can not transfer");
642   }
643
644
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 646

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
645   if (_getRateTokenAndETH() > 0) {
646   amountETH = (amount * 10 ** _decimals).div(_getRateTokenAndETH());
647   } else {
648   amountETH = 0;
649   }
650
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 646

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
645   if (_getRateTokenAndETH() > 0) {
646   amountETH = (amount * 10 ** _decimals).div(_getRateTokenAndETH());
647   } else {
648   amountETH = 0;
649   }
650
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 657

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
656   if (automatedMarketMakerPairs[from] && !_isExcludedMaxTransactionAmount[to]) {
657   require(amount + balanceOf(to) <= _maxwalletamount, "Max wallet exceeded");
658
659   } else if (automatedMarketMakerPairs[to] && !_isExcludedMaxTransactionAmount[from])
{
660   require(amountETH <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
661
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 664

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
663   require(amountETH <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");
664   require(amount + balanceOf(to) <= _maxwalletamount, "Max wallet exceeded");
665   } else if (_isExcludedMaxTransactionAmount[from] &&
!_isExcludedMaxTransactionAmount[to]) {
666   require(amount + balanceOf(to) <= _maxwalletamount, "Max wallet exceeded");
667   }
668
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 666

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
665   } else if (_isExcludedMaxTransactionAmount[from] &&
!_isExcludedMaxTransactionAmount[to]) {
666   require(amount + balanceOf(to) <= _maxwalletamount, "Max wallet exceeded");
667   }
668
669   if (!automatedMarketMakerPairs[from] && !_isExcludedMaxTransactionAmount[from]) {
670
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 670

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
669   if (!automatedMarketMakerPairs[from] && !_isExcludedMaxTransactionAmount[from]) {
670   if ( block.timestamp - transactionData[from].timeStamp < _dailyTimeLimit) {
671   require(transactionData[from].amount + amountETH <= _dailymaxTxAmount, "Transfer amount exceeds the dailymaxTxAmount.");
672   transactionData[from].amount += amountETH;
673   } else {
674
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 671

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
670   if ( block.timestamp - transactionData[from].timeStamp < _dailyTimeLimit) {
671   require(transactionData[from].amount + amountETH <= _dailymaxTxAmount, "Transfer
amount exceeds the dailymaxTxAmount.");
672   transactionData[from].amount += amountETH;
673   } else {
674   transactionData[from].timeStamp = block.timestamp;
675
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 672

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
671   require(transactionData[from].amount + amountETH <= _dailymaxTxAmount, "Transfer
amount exceeds the dailymaxTxAmount.");
672   transactionData[from].amount += amountETH;
673   } else {
674   transactionData[from].timeStamp = block.timestamp;
675   transactionData[from].amount = amountETH;
676
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 709

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
708   } else {
709   rateTokenAndETH = (amountToken * 10 ** _decimals).div(amountETH);
710   }
711   return rateTokenAndETH;
712   }
713
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 709

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
708   } else {
709   rateTokenAndETH = (amountToken * 10 ** _decimals).div(amountETH);
710   }
711   return rateTokenAndETH;
712   }
713
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 715

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
714   function swapAndLiquify(uint256 contractTokenBalance) internal lockTheSwap {
715   uint256 totalTokenAmount = _tokenAmountForFoundationTax +
_tokenAmountForLiquidityTax + _tokenAmountForMarketingTax + _tokenAmountForTreasuryTax;
716   uint256 tokenForMarketing = contractTokenBalance * (_tokenAmountForMarketingTax) /
totalTokenAmount;
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) /
totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax)
/ totalTokenAmount;
719
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 715

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
714   function swapAndLiquify(uint256 contractTokenBalance) internal lockTheSwap {
715   uint256 totalTokenAmount = _tokenAmountForFoundationTax +
_tokenAmountForLiquidityTax + _tokenAmountForMarketingTax + _tokenAmountForTreasuryTax;
716   uint256 tokenForMarketing = contractTokenBalance * (_tokenAmountForMarketingTax) /
totalTokenAmount;
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) /
totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax)
/ totalTokenAmount;
719
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 715

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
714   function swapAndLiquify(uint256 contractTokenBalance) internal lockTheSwap {
715   uint256 totalTokenAmount = _tokenAmountForFoundationTax +
_tokenAmountForLiquidityTax + _tokenAmountForMarketingTax + _tokenAmountForTreasuryTax;
716   uint256 tokenForMarketing = contractTokenBalance * (_tokenAmountForMarketingTax) /
totalTokenAmount;
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) /
totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax)
/ totalTokenAmount;
719
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 716

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
715   uint256 totalTokenAmount = _tokenAmountForFoundationTax + _tokenAmountForLiquidityTax + _tokenAmountForMarketingTax + _tokenAmountForTreasuryTax;
716   uint256 tokenForMarketing = contractTokenBalance * (_tokenAmountForMarketingTax) / totalTokenAmount;
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) / totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 716

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
715   uint256 totalTokenAmount = _tokenAmountForFoundationTax + _tokenAmountForLiquidityTax + _tokenAmountForMarketingTax + _tokenAmountForTreasuryTax;
716   uint256 tokenForMarketing = contractTokenBalance * (_tokenAmountForMarketingTax) / totalTokenAmount;
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) / totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 717

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
716   uint256 tokenForMarketing = contractTokenBalance * (_tokenAmountForMarketingTax) / totalTokenAmount;
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) / totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720   uint256 half = tokenForLiquidity.div(2);
721
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 717

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
716   uint256 tokenForMarketing = contractTokenBalance * (_tokenAmountForMarketingTax) / totalTokenAmount;
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) / totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720   uint256 half = tokenForLiquidity.div(2);
721
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 718

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) / totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720   uint256 half = tokenForLiquidity.div(2);
721   uint256 otherHalf = tokenForLiquidity.sub(half);
722
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 718

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
717   uint256 tokenForLiquidity = contractTokenBalance * (_tokenAmountForLiquidityTax) / totalTokenAmount;
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720   uint256 half = tokenForLiquidity.div(2);
721   uint256 otherHalf = tokenForLiquidity.sub(half);
722
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 719

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720   uint256 half = tokenForLiquidity.div(2);
721   uint256 otherHalf = tokenForLiquidity.sub(half);
722   uint256 initialBalance = address(this).balance;
723
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 719

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
718   uint256 tokenForFoundation = contractTokenBalance * (_tokenAmountForFoundationTax) / totalTokenAmount;
719   uint256 tokenForTreasury = contractTokenBalance * (_tokenAmountForTreasuryTax) / totalTokenAmount;
720   uint256 half = tokenForLiquidity.div(2);
721   uint256 otherHalf = tokenForLiquidity.sub(half);
722   uint256 initialBalance = address(this).balance;
723
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
722   uint256 initialBalance = address(this).balance;
723   uint256 tokenAmountForSwap = tokenForMarketing + tokenForFoundation +
tokenForTreasury + half;
724   swapTokensForEth(tokenAmountForSwap);
725
726   uint256 newBalance = address(this).balance.sub(initialBalance);
727
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
722   uint256 initialBalance = address(this).balance;
723   uint256 tokenAmountForSwap = tokenForMarketing + tokenForFoundation +
tokenForTreasury + half;
724   swapTokensForEth(tokenAmountForSwap);
725
726   uint256 newBalance = address(this).balance.sub(initialBalance);
727
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
722   uint256 initialBalance = address(this).balance;
723   uint256 tokenAmountForSwap = tokenForMarketing + tokenForFoundation +
tokenForTreasury + half;
724   swapTokensForEth(tokenAmountForSwap);
725
726   uint256 newBalance = address(this).balance.sub(initialBalance);
727
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 727

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
726   uint256 newBalance = address(this).balance.sub(initialBalance);
727   uint256 ethForLiquidity = newBalance * half / tokenAmountForSwap;
728   uint256 ethForMarketing = newBalance * tokenForMarketing / tokenAmountForSwap;
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 727

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
726   uint256 newBalance = address(this).balance.sub(initialBalance);
727   uint256 ethForLiquidity = newBalance * half / tokenAmountForSwap;
728   uint256 ethForMarketing = newBalance * tokenForMarketing / tokenAmountForSwap;
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 728

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
727   uint256 ethForLiquidity = newBalance * half / tokenAmountForSwap;
728   uint256 ethForMarketing = newBalance * tokenForMarketing / tokenAmountForSwap;
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
732
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 728

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
727   uint256 ethForLiquidity = newBalance * half / tokenAmountForSwap;
728   uint256 ethForMarketing = newBalance * tokenForMarketing / tokenAmountForSwap;
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
732
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 729

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
728   uint256 ethForMarketing = newBalance * tokenForMarketing / tokenAmountForSwap;
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
732   transferToAddressETH(payable(_marketingAddress), ethForMarketing.mul(90).div(100));
733
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 729

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
728   uint256 ethForMarketing = newBalance * tokenForMarketing / tokenAmountForSwap;
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
732   transferToAddressETH(payable(_marketingAddress), ethForMarketing.mul(90).div(100));
733
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 730

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
732   transferToAddressETH(payable(_marketingAddress), ethForMarketing.mul(90).div(100));
733   transferToAddressETH(payable(_treasuryAddress), ethForTreasury.mul(90).div(100));
734
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 730

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
729   uint256 ethForTreasury = newBalance * tokenForTreasury / tokenAmountForSwap;
730   uint256 ethForFoundation = newBalance * tokenForFoundation / tokenAmountForSwap;
731
732   transferToAddressETH(payable(_marketingAddress), ethForMarketing.mul(90).div(100));
733   transferToAddressETH(payable(_treasuryAddress), ethForTreasury.mul(90).div(100));
734
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 894

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
893   uint256 rLiquidity = tLiquidity.mul(currentRate);
894   _tokenAmountForLiquidityTax += tLiquidity;
895   _rOwned[address(this)] = _rOwned[address(this)].add(rLiquidity);
896   if(_isExcluded[address(this)])
897   _tOwned[address(this)] = _tOwned[address(this)].add(tLiquidity);
898
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 903

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
902   uint256 rMarketing = tMarketing.mul(currentRate);
903   _tokenAmountForMarketingTax += tMarketing;
904   _rOwned[address(this)] = _rOwned[address(this)].add(rMarketing);
905   if(_isExcluded[address(this)])
906   _tOwned[address(this)] = _tOwned[address(this)].add(tMarketing);
907
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 912

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
911   uint256 rTreasury = tTreasury.mul(currentRate);
912   _tokenAmountForTreasuryTax += tTreasury;
913   _rOwned[address(this)] = _rOwned[address(this)].add(rTreasury);
914   if(_isExcluded[address(this)])
915   _tOwned[address(this)] = _tOwned[address(this)].add(tTreasury);
916
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 921

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
920   uint256 rFoundation = tFoundation.mul(currentRate);
921   _tokenAmountForFoundationTax += tFoundation;
922   _rOwned[address(this)] = _rOwned[address(this)].add(rFoundation);
923   if(_isExcluded[address(this)])
924   _tOwned[address(this)] = _tOwned[address(this)].add(tFoundation);
925
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 961

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
960   uint256 tSupply = _tTotal;
961   for (uint256 i = 0; i < _excluded.length; i++) {
962   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
963   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
964   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
965
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 972

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
971   return _amount.mul(_reflectionTax).div(
972   10**2
973   );
974   }
975
976
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 978

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
977   return _amount.mul(_totalTax).div(
978   10**2
979   );
980   }
981
982
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1060

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1059   require(_isExcluded[account], "Account is already included");
1060   for (uint256 i = 0; i < _excluded.length; i++) {
1061   if (_excluded[i] == account) {
1062   _excluded[i] = _excluded[_excluded.length - 1];
1063   _tOwned[account] = 0;
1064
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1062

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1061   if (_excluded[i] == account) {
1062   _excluded[i] = _excluded[_excluded.length - 1];
1063   _tOwned[account] = 0;
1064   _isExcluded[account] = false;
1065   _excluded.pop();
1066
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1144

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1143   function setDailymaxTxAmountAndmaxTxAmount(uint256 dailymaxTxAmount, uint256 maxTxAmount) external onlyOwner() {
1144   _dailymaxTxAmount = dailymaxTxAmount * 10 ** _decimals;
1145   _maxTxAmount = maxTxAmount * 10 ** _decimals;
1146
1147   emit DaiyMaxTxAmountAndMaxTxAmountUpdate(dailymaxTxAmount, maxTxAmount);
1148
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1144

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1143   function setDailymaxTxAmountAndmaxTxAmount(uint256 dailymaxTxAmount, uint256 maxTxAmount) external onlyOwner() {
1144   _dailymaxTxAmount = dailymaxTxAmount * 10 ** _decimals;
1145   _maxTxAmount = maxTxAmount * 10 ** _decimals;
1146
1147   emit DaiyMaxTxAmountAndMaxTxAmountUpdate(dailymaxTxAmount, maxTxAmount);
1148
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1145

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1144   _dailymaxTxAmount = dailymaxTxAmount * 10 ** _decimals;
1145   _maxTxAmount = maxTxAmount * 10 ** _decimals;
1146
1147   emit DaiyMaxTxAmountAndMaxTxAmountUpdate(dailymaxTxAmount, maxTxAmount);
1148   }
1149
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1145

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1144   _dailymaxTxAmount = dailymaxTxAmount * 10 ** _decimals;
1145   _maxTxAmount = maxTxAmount * 10 ** _decimals;
1146
1147   emit DaiyMaxTxAmountAndMaxTxAmountUpdate(dailymaxTxAmount, maxTxAmount);
1148   }
1149
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1159

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1158   transactionData[account].lockedTime = block.timestamp;
1159   transactionData[account].lockPeriod = lockPeriod * 86400;
1160   emit LockAccount(account, true, lockPeriod);
1161   }
1162
1163
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1176

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1175   function airdrop(address recipient, uint256 amount) external onlyOwner() {
1176   _transfer(_msgSender(), recipient, amount * 10**18);
1177   }
1178
1179   function airdropInternal(address recipient, uint256 amount) internal {
1180
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1176

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1175   function airdrop(address recipient, uint256 amount) external onlyOwner() {
1176   _transfer(_msgSender(), recipient, amount * 10**18);
1177   }
1178
1179   function airdropInternal(address recipient, uint256 amount) internal {
1180
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1187

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1186   while(iterator < newholders.length){
1187   airdropInternal(newholders[iterator], amounts[iterator] * 10**18);
1188   iterator += 1;
1189   }
1190   }
1191
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1187

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1186   while(iterator < newholders.length){
1187   airdropInternal(newholders[iterator], amounts[iterator] * 10**18);
1188   iterator += 1;
1189   }
1190   }
1191
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1188

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1187   airdropInternal(newholders[iterator], amounts[iterator] * 10**18);
1188   iterator += 1;
1189   }
1190   }
1191
1192
```

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

LINE 1062

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1061   if (_excluded[i] == account) {
1062   _excluded[i] = _excluded[_excluded.length - 1];
1063   _tOwned[account] = 0;
1064   _isExcluded[account] = false;
1065   _excluded.pop();
1066
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 429

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
428
429   bool _tradingActive = true;
430   bool inSwapAndLiquify;
431   bool public swapAndLiquifyEnabled = true;
432
433
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 430

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

- ArhcieNeko.sol

Locations

```
429   bool _tradingActive = true;
430   bool inSwapAndLiquify;
431   bool public swapAndLiquifyEnabled = true;
432
433   string private _name = "Archie Neko";
434
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 749

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
748   address[] memory path = new address[](2);
749   path[0] = address(this);
750   path[1] = uniswapV2Router.WETH();
751
752   _approve(address(this), address(uniswapV2Router), tokenAmount);
753
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 750

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
749   path[0] = address(this);
750   path[1] = uniswapV2Router.WETH();
751
752   _approve(address(this), address(uniswapV2Router), tokenAmount);
753
754
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 962

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
961   for (uint256 i = 0; i < _excluded.length; i++) {
962   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
963   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
964   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
965   }
966
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 962

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
961   for (uint256 i = 0; i < _excluded.length; i++) {
962   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
963   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
964   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
965   }
966
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 963

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
962   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
963   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
964   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
965   }
966   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
967
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 964

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
963   rSupply = rSupply.sub(_rOwned[_excluded[i]]);
964   tSupply = tSupply.sub(_tOwned[_excluded[i]]);
965   }
966   if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
967   return (rSupply, tSupply);
968
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1061

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1060   for (uint256 i = 0; i < _excluded.length; i++) {
1061   if (_excluded[i] == account) {
1062   _excluded[i] = _excluded[_excluded.length - 1];
1063   _tOwned[account] = 0;
1064   _isExcluded[account] = false;
1065
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1062

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1061   if (_excluded[i] == account) {
1062   _excluded[i] = _excluded[_excluded.length - 1];
1063   _tOwned[account] = 0;
1064   _isExcluded[account] = false;
1065   _excluded.pop();
1066
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1062

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1061   if (_excluded[i] == account) {
1062   _excluded[i] = _excluded[_excluded.length - 1];
1063   _tOwned[account] = 0;
1064   _isExcluded[account] = false;
1065   _excluded.pop();
1066
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1187

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1186   while(iterator < newholders.length){
1187   airdropInternal(newholders[iterator], amounts[iterator] * 10**18);
1188   iterator += 1;
1189   }
1190   }
1191
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1187

low SEVERITY

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

Source File

- ArhcieNeko.sol

Locations

```
1186   while(iterator < newholders.length){
1187   airdropInternal(newholders[iterator], amounts[iterator] * 10**18);
1188   iterator += 1;
1189   }
1190   }
1191
```

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.