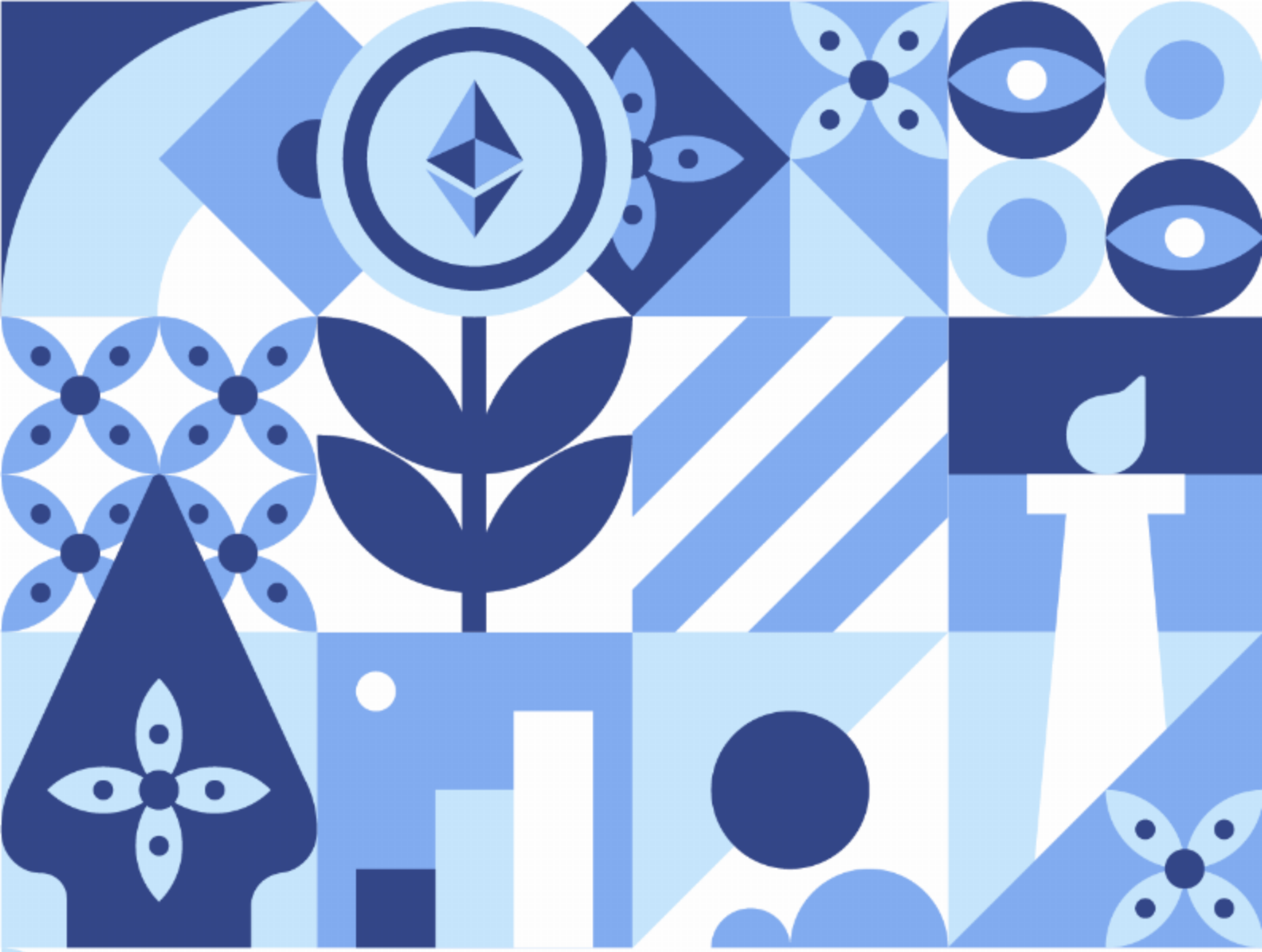

Half Floki

Smart Contract Audit Report

SYSFIXED

09 Feb 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Half Floki	FLOKI0.5	Ethereum

Addresses

Contract address	0x05cb9bfD7995B0bA426c938225A89bDC6b560fCB
Contract deployer address	0x83ce315137aa2dc26784E82A88292ADAdB4db7Ac

Project Website

http://halffloki.net/

Codebase

https://etherscan.io/address/0x05cb9bfD7995B0bA426c938225A89bDC6b560fCB#code

SUMMARY

Half Floki $floki0.5 is a decentralized cryptocurrency built to be more accessible and equitable. It uses a unique halving process that splits each Floki coin into two halves, known as "Minors" and "Majors".

Contract Summary

Documentation Quality

Half 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 Half Floki with the discovery of several low issues.

Test Coverage

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

Audit Findings Summary

- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 343, 376, 378, 399, 400, 496, 510, 525, 526, 539, 551, 566, 580, 594, 608, 624, 647, 670, 696, 785, 787, 787, 788, 788, 789, 789, 793, 797, 838, 838, 842, 842, 851, 851, 851, 854, 854, 859, 859, 859, 862, 862, 883, 893, 954, 960, 1001, 1001, 1002, 1002, 1007, 1007, 1008, 1008, 1018, 1048 and 1049.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 1.
- 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 1027 and 1028.

CONCLUSION

We have audited the Half Floki project released on February 2023 to discover issues and identify potential security vulnerabilities in Half 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 a satisfactory result with some low-risk issues.

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Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.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

DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS
Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	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	Wednesday Feb 08 2023 23:55:48 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Feb 09 2023 17:55:51 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	HalfFLOKI.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-103	NO PRAGMA IS SET.	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 343

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
342   unchecked {
343   _approve(sender, _msgSender(), currentAllowance - amount);
344   }
345
346   return true;
347
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 376

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
375   unchecked {
376   _balances[sender] = senderBalance - amount;
377   }
378   _balances[recipient] += amount;
379
380
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 378

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
377   }
378   _balances[recipient] += amount;
379
380   emit Transfer(sender, recipient, amount);
381
382
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 399

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
398
399   _totalSupply += amount;
400   _balances[account] += amount;
401   emit Transfer(address(0), account, amount);
402
403
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 400

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
399   _totalSupply += amount;
400   _balances[account] += amount;
401   emit Transfer(address(0), account, amount);
402
403   _afterTokenTransfer(address(0), account, amount);
404
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 496

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
495   unchecked {
496   uint256 c = a + b;
497   if (c < a) return (false, 0);
498   return (true, c);
499   }
500
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 510

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
509   if (b > a) return (false, 0);
510   return (true, a - b);
511   }
512   }
513
514
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 525

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
524   if (a == 0) return (true, 0);
525   uint256 c = a * b;
526   if (c / a != b) return (false, 0);
527   return (true, c);
528   }
529
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 526

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
525   uint256 c = a * b;
526   if (c / a != b) return (false, 0);
527   return (true, c);
528   }
529   }
530
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 539

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
538   if (b == 0) return (false, 0);
539   return (true, a / b);
540   }
541   }
542
543
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 551

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
550   if (b == 0) return (false, 0);
551   return (true, a % b);
552   }
553   }
554
555
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 566

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
565   function add(uint256 a, uint256 b) internal pure returns (uint256) {
566   return a + b;
567   }
568
569   /**
570
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 580

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
579   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
580   return a - b;
581   }
582
583   /**
584
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 594

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
593   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
594   return a * b;
595   }
596
597   /**
598
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 608

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
607   function div(uint256 a, uint256 b) internal pure returns (uint256) {
608   return a / b;
609   }
610
611   /**
612
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 624

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
623   function mod(uint256 a, uint256 b) internal pure returns (uint256) {
624   return a % b;
625   }
626
627   /**
628
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 647

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
646   require(b <= a, errorMessage);
647   return a - b;
648   }
649   }
650
651
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 670

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
669   require(b > 0, errorMessage);
670   return a / b;
671   }
672   }
673
674
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 696

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
695   require(b > 0, errorMessage);
696   return a % b;
697   }
698   }
699   }
700
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 785

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
784
785   uint256 totalSupply = 1_000_000 * 1e18;
786
787   maxTransactionAmount =  totalSupply * 2 / 100; // 2% from total supply
maxTransactionAmountTxn
788   maxWallet = totalSupply * 2 / 100; // 2% from total supply maxWallet
789
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 787

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
786
787   maxTransactionAmount =  totalSupply * 2 / 100; // 2% from total supply
maxTransactionAmountTxn
788   maxWallet = totalSupply * 2 / 100; // 2% from total supply maxWallet
789   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
790
791
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 787

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
786
787   maxTransactionAmount =  totalSupply * 2 / 100; // 2% from total supply
maxTransactionAmountTxn
788   maxWallet = totalSupply * 2 / 100; // 2% from total supply maxWallet
789   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
790
791
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 788

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
787   maxTransactionAmount =  totalSupply * 2 / 100; // 2% from total supply
maxTransactionAmountTxn
788   maxWallet = totalSupply * 2 / 100; // 2% from total supply maxWallet
789   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
790
791   buyDevFee = _buyDevFee;
792
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 788

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
787   maxTransactionAmount =  totalSupply * 2 / 100; // 2% from total supply
maxTransactionAmountTxn
788   maxWallet = totalSupply * 2 / 100; // 2% from total supply maxWallet
789   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
790
791   buyDevFee = _buyDevFee;
792
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 789

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
788   maxWallet = totalSupply * 2 / 100; // 2% from total supply maxWallet
789   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
790
791   buyDevFee = _buyDevFee;
792   buyLiquidityFee = _buyLiquidityFee;
793
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 789

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
788   maxWallet = totalSupply * 2 / 100; // 2% from total supply maxWallet
789   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
790
791   buyDevFee = _buyDevFee;
792   buyLiquidityFee = _buyLiquidityFee;
793
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 793

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
792   buyLiquidityFee = _buyLiquidityFee;
793   buyTotalFees = buyDevFee + buyLiquidityFee;
794
795   sellDevFee = _sellDevFee;
796   sellLiquidityFee = _sellLiquidityFee;
797
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 797

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
796   sellLiquidityFee = _sellLiquidityFee;
797   sellTotalFees = sellDevFee + sellLiquidityFee;
798
799   devWallet = address(0x83ce315137aa2dc26784E82A88292ADAdB4db7Ac);
800
801
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 838

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
837   require(
838   newAmount >= (totalSupply() * 1) / 100000,
839   "Swap amount cannot be lower than 0.001% total supply."
840   );
841   require(
842
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 838

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
837   require(
838   newAmount >= (totalSupply() * 1) / 100000,
839   "Swap amount cannot be lower than 0.001% total supply."
840   );
841   require(
842
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 842

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
841   require(
842   newAmount <= (totalSupply() * 5) / 1000,
843   "Swap amount cannot be higher than 0.5% total supply."
844   );
845   swapTokensAtAmount = newAmount;
846
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 842

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
841   require(
842   newAmount <= (totalSupply() * 5) / 1000,
843   "Swap amount cannot be higher than 0.5% total supply."
844   );
845   swapTokensAtAmount = newAmount;
846
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 851

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
850   require(
851   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
852   "Cannot set maxTransactionAmount lower than 0.1%"
853   );
854   maxTransactionAmount = newNum * (10**18);
855
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 851

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
850   require(
851   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
852   "Cannot set maxTransactionAmount lower than 0.1%"
853   );
854   maxTransactionAmount = newNum * (10**18);
855
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 851

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
850   require(
851   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
852   "Cannot set maxTransactionAmount lower than 0.1%"
853   );
854   maxTransactionAmount = newNum * (10**18);
855
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 854

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
853   );
854   maxTransactionAmount = newNum * (10**18);
855   }
856
857   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
858
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 854

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
853   );
854   maxTransactionAmount = newNum * (10**18);
855   }
856
857   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
858
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 859

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
858   require(
859   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
860   "Cannot set maxWallet lower than 0.5%"
861   );
862   maxWallet = newNum * (10**18);
863
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 859

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
858   require(
859   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
860   "Cannot set maxWallet lower than 0.5%"
861   );
862   maxWallet = newNum * (10**18);
863
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 859

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
858   require(
859   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
860   "Cannot set maxWallet lower than 0.5%"
861   );
862   maxWallet = newNum * (10**18);
863
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 862

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
861   );
862   maxWallet = newNum * (10**18);
863   }
864
865   function excludeFromMaxTransaction(address updAds, bool isEx)
866
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 862

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
861   );
862   maxWallet = newNum * (10**18);
863   }
864
865   function excludeFromMaxTransaction(address updAds, bool isEx)
866
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 883

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
882   buyLiquidityFee = _liquidityFee;
883   buyTotalFees = buyDevFee + buyLiquidityFee;
884   require(buyTotalFees <= 10, "Must keep fees at 10% or less");
885   }
886
887
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 893

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
892   sellLiquidityFee = _liquidityFee;
893   sellTotalFees = sellDevFee + sellLiquidityFee;
894   require(sellTotalFees <= 99, "Must keep fees at 15% or less");
895   }
896
897
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 954

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
953   require(
954   amount + balanceOf(to) <= maxWallet,
955   "Max wallet exceeded"
956   );
957   }
958
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 960

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
959   require(
960   amount + balanceOf(to) <= maxWallet,
961   "Max wallet exceeded"
962   );
963   }
964
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1001

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1000   fees = amount.mul(sellTotalFees).div(100);
1001   tokensForLiquidity = (fees * sellLiquidityFee) / sellTotalFees;
1002   tokensForDev = (fees * sellDevFee) / sellTotalFees;
1003   }
1004   // on buy
1005
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1001

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1000   fees = amount.mul(sellTotalFees).div(100);
1001   tokensForLiquidity = (fees * sellLiquidityFee) / sellTotalFees;
1002   tokensForDev = (fees * sellDevFee) / sellTotalFees;
1003   }
1004   // on buy
1005
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1002

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1001   tokensForLiquidity = (fees * sellLiquidityFee) / sellTotalFees;
1002   tokensForDev = (fees * sellDevFee) / sellTotalFees;
1003   }
1004   // on buy
1005   else if (from == uniswapV2Pair && buyTotalFees > 0) {
1006
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1002

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1001   tokensForLiquidity = (fees * sellLiquidityFee) / sellTotalFees;
1002   tokensForDev = (fees * sellDevFee) / sellTotalFees;
1003   }
1004   // on buy
1005   else if (from == uniswapV2Pair && buyTotalFees > 0) {
1006
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1007

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1006   fees = amount.mul(buyTotalFees).div(100);
1007   tokensForLiquidity = (fees * buyLiquidityFee) / buyTotalFees;
1008   tokensForDev = (fees * buyDevFee) / buyTotalFees;
1009   }
1010
1011
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1007

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1006   fees = amount.mul(buyTotalFees).div(100);
1007   tokensForLiquidity = (fees * buyLiquidityFee) / buyTotalFees;
1008   tokensForDev = (fees * buyDevFee) / buyTotalFees;
1009   }
1010
1011
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1008

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1007   tokensForLiquidity = (fees * buyLiquidityFee) / buyTotalFees;
1008   tokensForDev = (fees * buyDevFee) / buyTotalFees;
1009   }
1010
1011   if (fees> 0) {
1012
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1008

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1007   tokensForLiquidity = (fees * buyLiquidityFee) / buyTotalFees;
1008   tokensForDev = (fees * buyDevFee) / buyTotalFees;
1009   }
1010
1011   if (fees> 0) {
1012
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1018

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1017
1018   amount -= fees;
1019   }
1020
1021   super._transfer(from, to, amount);
1022
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1048

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1047
1048   if (contractBalance > swapTokensAtAmount * 20) {
1049   contractBalance = swapTokensAtAmount * 20;
1050   }
1051
1052
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1049

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1048   if (contractBalance > swapTokensAtAmount * 20) {
1049   contractBalance = swapTokensAtAmount * 20;
1050   }
1051
1052   swapTokensForFLOKI(contractBalance);
1053
```

SWC-103 | NO PRAGMA IS SET.

LINE 1

low SEVERITY

It is recommended to make a conscious choice on what version of Solidity is used for compilation. Currently no version is set in the Solidity file.

Source File

- HalfFLOKI.sol

Locations

```
0
1   //https://t.me/FlokiHalfETH
2
3   abstract contract Context {
4   function _msgSender() internal view virtual returns (address) {
5
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1027

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1026   address[] memory path = new address[](2);
1027   path[0] = address(this);
1028   path[1] = FLOKI;
1029
1030   _approve(address(this), address(uniswapV2Router), tokenAmount);
1031
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1028

low SEVERITY

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

Source File

- HalfFLOKI.sol

Locations

```
1027   path[0] = address(this);
1028   path[1] = FLOKI;
1029
1030   _approve(address(this), address(uniswapV2Router), tokenAmount);
1031
1032
```

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.