FUFTX

Smart Contract Audit Report

SYSFIXED

12 Dec 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
FUFTX	FUFTX	Binance Smart Chain

Addresses

Contract address	0xF27F5F369FbBc7716f51ad34C4050801D38DB151
Contract deployer address	0x331226F3F50AE68910030CCbcA8248f74a5816c8

Project Website

https://www.fuftx.com/

Codebase

https://bscscan.com/address/0xF27F5F369FbBc7716f51ad34C4050801D38DB151#code

SUMMARY

Created by the founders of some of the largest, most successful projects on BNB Chain (most of which have grown over 5000%). FU FTX is a BNB Chain token on a mission to give FTX the biggest F U in history for what they did to everyone - all while helping support victims of the FTX collapse. 8% tax, 4% to victim relief, 4% to community/admin.

Contract Summary

Documentation Quality

FUFTX 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 FUFTX 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 45, 54, 61, 62, 70, 77, 82, 86, 90, 94, 98, 108, 119, 130, 349, 350, 354, 355, 356, 475, 494, 519, 534, 536, 561, 578, 579, 583, 585, 587, 592, 596, 598, 602, 606, 608, 612, 616, 618, 636, 637, 638, 641, 642, 643, 649, 655, 656, 657, 658, 659, 660, 666, 672, 674, 675, 677, 707, 722, 742, 745, 748, 749, 756, 756, 784, 791, 792, 796, 850, 851, 855, 872 and 536.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 15.
- SWC-110 | It is recommended to use revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 535, 536, 673, 674, 675, 770, 771, 820 and 821.

CONCLUSION

We have audited the FUFTX project which has released on December 2022 to discover issues and identify potential security vulnerabilities in FUFTX Project. This process is used to find technical issues and security loopholes that find some common issues in the code.

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

The most common issue found in writing code on contracts that do not pose a big risk, writing on contracts is close to the standard of writing contracts in general. The low-level 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 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
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
Assert Violation	SWC-110	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 Caller	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
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
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

SMART CONTRACT ANALYSIS

Started	Sunday Dec 11 2022 20:59:42 GMT+0000 (Coordinated Universal Time)
Finished	Monday Dec 12 2022 11:17:43 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	FUFTX.sol

Detected Issues

ID	Title	Severity	Status
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "%" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "%" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "%" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "%" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "++" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "++" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "-" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged

SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "**" DISCOVERED	low	acknowledged
SWC-101	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-103	A FLOATING PRAGMA IS 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 45

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
44   unchecked {
45   uint256 c = a + b;
46   if (c < a) return (false, 0);
47   return (true, c);
48   }
49
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 54

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
53   if (b > a) return (false, 0);
54   return (true, a - b);
55   }
56   }
57
58
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 61

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
60   if (a == 0) return (true, 0);
61   uint256 c = a * b;
62   if (c / a != b) return (false, 0);
63   return (true, c);
64   }
65
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 62

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
61   uint256 c = a * b;
62   if (c / a != b) return (false, 0);
63   return (true, c);
64   }
65   }
66
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 70

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
69   if (b == 0) return (false, 0);
70   return (true, a / b);
71   }
72   }
73
74
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 77

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
76   if (b == 0) return (false, 0);
77   return (true, a % b);
78   }
79   }
80
81
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 82

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
81   function add(uint256 a, uint256 b) internal pure returns (uint256) {
82   return a + b;
83   }
84
85   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
86
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 86

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
85   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
86   return a - b;
87   }
88
89   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
90
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 90

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
89   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
90   return a * b;
91   }
92
93   function div(uint256 a, uint256 b) internal pure returns (uint256) {
94
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 94

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
93   function div(uint256 a, uint256 b) internal pure returns (uint256) {
94   return a / b;
95   }
96
97   function mod(uint256 a, uint256 b) internal pure returns (uint256) {
98
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 98

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
97   function mod(uint256 a, uint256 b) internal pure returns (uint256) {
98   return a % b;
99   }
100
101   function sub(
102
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 108

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
107   require(b <= a, errorMessage);
108   return a - b;
109   }
110   }
111
112
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 119

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
118   require(b > 0, errorMessage);
119   return a / b;
120   }
121   }
122
123
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 130

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
129   require(b > 0, errorMessage);
130   return a % b;
131   }
132   }
133   }
134
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 349

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
348
349   uint256 private _tTotal = 1000000000 * 10**_decimals;
350   uint256 private _rTotal = (MAX - (MAX % _tTotal));
351
352   uint256 public maxBuyAmount = _tTotal.mul(2).div(100);
353
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 350

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
349   uint256 private _tTotal = 1000000000 * 10**_decimals;
350   uint256 private _rTotal = (MAX - (MAX % _tTotal));
351
352   uint256 public maxBuyAmount = _tTotal.mul(2).div(100);
353   uint256 public maxSellAmount = _tTotal.mul(1).div(100);
354
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 350

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
349   uint256 private _tTotal = 1000000000 * 10**_decimals;
350   uint256 private _rTotal = (MAX - (MAX % _tTotal));
351
352   uint256 public maxBuyAmount = _tTotal.mul(2).div(100);
353   uint256 public maxSellAmount = _tTotal.mul(1).div(100);
354
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 354

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
353   uint256 public maxSellAmount = _tTotal.mul(1).div(100);
354   uint256 public swapTokensAtAmount = 500000000000 * 10**_decimals;
355   uint256 public _maxWalletSize = 20000000000000 * 10**9;
356   uint256 public buyBackUpperLimit = 1 * 10**18;
357
358
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 355

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
354   uint256 public swapTokensAtAmount = 500000000000 * 10**_decimals;
355   uint256 public _maxWalletSize = 20000000000000 * 10**9;
356   uint256 public buyBackUpperLimit = 1 * 10**18;
357
358   address public FUAddress = 0xF753ab8d6F9c29CF7E440179DCb8DBe6A68e8e39;
359
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 356

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
355   uint256 public _maxWalletSize = 20000000000000 * 10**9;
356   uint256 public buyBackUpperLimit = 1 * 10**18;
357
358   address public FUAddress = 0xF753ab8d6F9c29CF7E440179DCb8DBe6A68e8e39;
359   address public operationsAddress = 0xBA12b05C5635B4c0Dd888F7748FdA1D83CBA8E35;
360
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 475

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
474   function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
475   _approve(_msgSender(), spender, _allowances[_msgSender()][spender]+addedValue);
476   return true;
477   }
478
479
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 494

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
493   _rTotal = _rTotal.sub(s.rAmount);
494   totFeesPaid.rfi += tAmount;
495   }
496
497
498
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 519

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
518   uint256 currentRate =  _getRate();
519   return rAmount/currentRate;
520   }
521
522   //@dev kept original RFI naming -> "reward" as in reflection
523
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 534

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
533   require(_isExcluded[account], "Account is not excluded");
534   for (uint256 i = 0; i < _excluded.length; i++) {
535   if (_excluded[i] == account) {
536   _excluded[i] = _excluded[_excluded.length - 1];
537   _tOwned[account] = 0;
538
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 536

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
535   if (_excluded[i] == account) {
536   _excluded[i] = _excluded[_excluded.length - 1];
537   _tOwned[account] = 0;
538   _isExcluded[account] = false;
539   _excluded.pop();
540
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 561

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
560   _maxWalletSize = _tTotal.mul(maxWallPercent).div(
561   10**2
562   );
563   }
564
565
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 578

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
577   function _reflectRfi(uint256 rRfi, uint256 tRfi) private {
578   _rTotal -=rRfi;
579   totFeesPaid.rfi +=tRfi;
580   }
581
582
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 579

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
578   _rTotal -=rRfi;
579   totFeesPaid.rfi +=tRfi;
580   }
581
582   function _takeOperations(uint256 rOperations, uint256 tOperations) private {
583
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 583

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
582   function _takeOperations(uint256 rOperations, uint256 tOperations) private {
583   totFeesPaid.operations +=tOperations;
584   if(_isExcluded[address(this)]){
585   _tOwned[address(this)]+=tOperations;
586   }
587
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 585

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
584   if(_isExcluded[address(this)]){
585   _tOwned[address(this)]+=tOperations;
586   }
587   _rOwned[address(this)] +=rOperations;
588
589
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 587

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
586   }
587   _rOwned[address(this)] +=rOperations;
588
589   }
590
591
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 592

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
591   function _takeBuyback(uint256 rBuyback, uint256 tBuyback) private {
592   totFeesPaid.buyBack +=tBuyback;
593
594   if(_isExcluded[address(this)])
595   {
596
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 596

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
595   {
596   _tOwned[address(this)]+=tBuyback;
597   }
598   _rOwned[address(this)] +=rBuyback;
599   }
600
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 598

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
597   }
598   _rOwned[address(this)] +=rBuyback;
599   }
600
601   function _takeLiquidity(uint256 rLiquidity, uint256 tLiquidity) private {
602
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 602

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
601   function _takeLiquidity(uint256 rLiquidity, uint256 tLiquidity) private {
602   totFeesPaid.liquidity +=tLiquidity;
603
604   if(_isExcluded[address(this)])
605   {
606
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 606

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
605   {
606   _tOwned[address(this)]+=tLiquidity;
607   }
608   _rOwned[address(this)] +=rLiquidity;
609   }
610
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 608

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
607   }
608   _rOwned[address(this)] +=rLiquidity;
609   }
610
611   function _takeFU(uint256 rFU, uint256 tFU) private {
612
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 612

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
611   function _takeFU(uint256 rFU, uint256 tFU) private {
612   totFeesPaid.FU +=tFU;
613
614   if(_isExcluded[FUAddress])
615   {
616
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 616

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
615   {
616   _tOwned[FUAddress]+=tFU;
617   }
618   _rOwned[FUAddress] +=rFU;
619   }
620
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 618

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
617   }
618   _rOwned[FUAddress] +=rFU;
619   }
620
621
622
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 636

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
635   if(isSale){
636   s.tOperations = tAmount*sellFeeRates.operations/1000;
637   s.tFU = tAmount*sellFeeRates.FU/1000;
638   s.tTransferAmount = tAmount-s.tRfi-s.tOperations-s.tFU-s.tLiquidity-s.tBuyback;
639   }
640
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 637

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
636   s.tOperations = tAmount*sellFeeRates.operations/1000;
637   s.tFU = tAmount*sellFeeRates.FU/1000;
638   s.tTransferAmount = tAmount-s.tRfi-s.tOperations-s.tFU-s.tLiquidity-s.tBuyback;
639   }
640   else{
641
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 638

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
637   s.tFU = tAmount*sellFeeRates.FU/1000;
638   s.tTransferAmount = tAmount-s.tRfi-s.tOperations-s.tFU-s.tLiquidity-s.tBuyback;
639   }
640   else{
641   s.tOperations = tAmount*feeRates.operations/1000;
642
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 641

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
640   else{
641   s.tOperations = tAmount*feeRates.operations/1000;
642   s.tFU = tAmount*feeRates.FU/1000;
643   s.tTransferAmount = tAmount-s.tRfi-s.tOperations-s.tFU-s.tLiquidity-s.tBuyback;
644   }
645
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 642

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
641   s.tOperations = tAmount*feeRates.operations/1000;
642   s.tFU = tAmount*feeRates.FU/1000;
643   s.tTransferAmount = tAmount-s.tRfi-s.tOperations-s.tFU-s.tLiquidity-s.tBuyback;
644   }
645   return s;
646
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 643

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
642   s.tFU = tAmount*feeRates.FU/1000;
643   s.tTransferAmount = tAmount-s.tRfi-s.tOperations-s.tFU-s.tLiquidity-s.tBuyback;
644   }
645   return s;
646   }
647
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 649

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
648   function _getRValues(valuesFromGetValues memory s, uint256 tAmount, bool takeFee,
uint256 currentRate) private pure returns (uint256 rAmount, uint256 rTransferAmount,
uint256 rRfi, uint256 rOperations,uint256 rFU, uint256 rLiquidity, uint256 rBuyback) {
649   rAmount = tAmount*currentRate;
650
651   if(!takeFee) {
652   return(rAmount, rAmount, 0,0,0,0,0);
653
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 655

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
654
655   rRfi = s.tRfi*currentRate;
656   rOperations = s.tOperations*currentRate;
657   rFU = s.tFU*currentRate;
658   rLiquidity = s.tLiquidity*currentRate;
659
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 656

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
655   rRfi = s.tRfi*currentRate;
656   rOperations = s.tOperations*currentRate;
657   rFU = s.tFU*currentRate;
658   rLiquidity = s.tLiquidity*currentRate;
659   rBuyback = s.tBuyback*currentRate;
660
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 657

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
656   rOperations = s.tOperations*currentRate;
657   rFU = s.tFU*currentRate;
658   rLiquidity = s.tLiquidity*currentRate;
659   rBuyback = s.tBuyback*currentRate;
660   rTransferAmount =  rAmount-rRfi-rOperations-rFU-rLiquidity-rBuyback;
661
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 658

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
657   rFU = s.tFU*currentRate;
658   rLiquidity = s.tLiquidity*currentRate;
659   rBuyback = s.tBuyback*currentRate;
660   rTransferAmount =  rAmount-rRfi-rOperations-rFU-rLiquidity-rBuyback;
661   return (rAmount, rTransferAmount, rRfi,rOperations,rFU,rLiquidity, rBuyback);
662
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 659

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
658   rLiquidity = s.tLiquidity*currentRate;
659   rBuyback = s.tBuyback*currentRate;
660   rTransferAmount =  rAmount-rRfi-rOperations-rFU-rLiquidity-rBuyback;
661   return (rAmount, rTransferAmount, rRfi,rOperations,rFU,rLiquidity, rBuyback);
662   }
663
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 660

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
659   rBuyback = s.tBuyback*currentRate;
660   rTransferAmount =  rAmount-rRfi-rOperations-rFU-rLiquidity-rBuyback;
661   return (rAmount, rTransferAmount, rRfi,rOperations,rFU,rLiquidity, rBuyback);
662   }
663
664
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 666

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
665   (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
666   return rSupply/tSupply;
667   }
668
669   function _getCurrentSupply() private view returns(uint256, uint256) {
670
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 672

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
671   uint256 tSupply = _tTotal;
672   for (uint256 i = 0; i < _excluded.length; i++) {
673   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
674   rSupply = rSupply-_rOwned[_excluded[i]];
675   tSupply = tSupply-_tOwned[_excluded[i]];
676
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 674

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
673   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
674   rSupply = rSupply-_rOwned[_excluded[i]];
675   tSupply = tSupply-_tOwned[_excluded[i]];
676   }
677   if (rSupply < _rTotal/_tTotal) return (_rTotal, _tTotal);
678
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 675

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
674   rSupply = rSupply-_rOwned[_excluded[i]];
675   tSupply = tSupply-_tOwned[_excluded[i]];
676   }
677   if (rSupply < _rTotal/_tTotal) return (_rTotal, _tTotal);
678   return (rSupply, tSupply);
679
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 677

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
676   }
677   if (rSupply < _rTotal/_tTotal) return (_rTotal, _tTotal);
678   return (rSupply, tSupply);
679   }
680
681
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 707

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
706   uint256 walletCurrentBalance = balanceOf(to);
707   require(walletCurrentBalance + amount <= _maxWalletSize, "Exceeds maximum wallet
token amount");
708   }
709
710   if( from != owner() &&
711
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 722

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
721   uint256 balance = address(this).balance;
722   if (buyBackEnabled && balance > uint256(1 * 10**18) && to == pair) {
723   if (balance > buyBackUpperLimit) balance = buyBackUpperLimit;
724   buyBackTokens(balance.div(100));
725   }
726
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 742

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
741   if (_isExcluded[sender] ) {  //from excluded
742   _tOwned[sender] = _tOwned[sender]-tAmount;
743   }
744   if (_isExcluded[recipient]) { //to excluded
745   _tOwned[recipient] = _tOwned[recipient]+s.tTransferAmount;
746
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 745

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
744   if (_isExcluded[recipient]) { //to excluded
745   _tOwned[recipient] = _tOwned[recipient]+s.tTransferAmount;
746   }
747
748   _rOwned[sender] = _rOwned[sender]-s.rAmount;
749
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 748

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
747
748   _rOwned[sender] = _rOwned[sender]-s.rAmount;
749   _rOwned[recipient] = _rOwned[recipient]+s.rTransferAmount;
750   _reflectRfi(s.rRfi, s.tRfi);
751   _takeOperations(s.rOperations,s.tOperations);
752
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 749

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
748   _rOwned[sender] = _rOwned[sender]-s.rAmount;
749   _rOwned[recipient] = _rOwned[recipient]+s.rTransferAmount;
750   _reflectRfi(s.rRfi, s.tRfi);
751   _takeOperations(s.rOperations,s.tOperations);
752   _takeLiquidity(s.rLiquidity,s.tLiquidity);
753
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 756

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
755   emit Transfer(sender, recipient, s.tTransferAmount);
756   emit Transfer(sender, address(this), s.tLiquidity + s.tOperations + s.tBuyback);
757   emit Transfer(sender, FUAddress, s.tFU);
758
759   }
760
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 756

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
755   emit Transfer(sender, recipient, s.tTransferAmount);
756   emit Transfer(sender, address(this), s.tLiquidity + s.tOperations + s.tBuyback);
757   emit Transfer(sender, FUAddress, s.tFU);
758
759   }
760
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 784

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
783   // Split the contract balance into halves
784   uint256 denominator= (feeRates.operations) * 2;
785   uint256 toSwap = tokens;
786
787   uint256 initialBalance = address(this).balance;
788
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 791

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
790
791   uint256 deltaBalance = address(this).balance - initialBalance;
792   uint256 unitBalance= deltaBalance / (denominator);
793
794
795
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 792

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
791   uint256 deltaBalance = address(this).balance - initialBalance;
792   uint256 unitBalance= deltaBalance / (denominator);
793
794
795   // Send BNB to operationsWallet
796
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 796

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
795   // Send BNB to operationsWallet
796   uint256 operationsAmt = unitBalance * 2 * feeRates.operations;
797   if(operationsAmt > 0){
798   payable(operationsAddress).transfer(operationsAmt);
799   }
800
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 796

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
795   // Send BNB to operationsWallet
796   uint256 operationsAmt = unitBalance * 2 * feeRates.operations;
797   if(operationsAmt > 0){
798   payable(operationsAddress).transfer(operationsAmt);
799   }
800
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 850

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
849   function setMaxBuyAndSellAmount(uint256 _maxBuyamount, uint256 _maxSellAmount)
external onlyOwner{
850   maxBuyAmount = _maxBuyamount * 10**9;
851   maxSellAmount = _maxSellAmount * 10**9;
852   }
853
854
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 850

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
849   function setMaxBuyAndSellAmount(uint256 _maxBuyamount, uint256 _maxSellAmount)
external onlyOwner{
850   maxBuyAmount = _maxBuyamount * 10**9;
851   maxSellAmount = _maxSellAmount * 10**9;
852   }
853
854
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 851

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
850   maxBuyAmount = _maxBuyamount * 10**9;
851   maxSellAmount = _maxSellAmount * 10**9;
852   }
853
854   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
855
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 851

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
850   maxBuyAmount = _maxBuyamount * 10**9;
851   maxSellAmount = _maxSellAmount * 10**9;
852   }
853
854   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
855
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 855

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
854   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
855   swapTokensAtAmount = amount * 10**_decimals;
856   }
857
858   function updateSwapEnabled(bool _enabled) external onlyOwner{
859
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 855

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
854   function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
855   swapTokensAtAmount = amount * 10**_decimals;
856   }
857
858   function updateSwapEnabled(bool _enabled) external onlyOwner{
859
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 872

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
871   function setBuybackUpperLimit(uint256 buyBackLimit) external onlyOwner() {
872   buyBackUpperLimit = buyBackLimit * 10**15;
873   }
874
875   function isBot(address account) public view returns(bool){
876
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 872

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
871   function setBuybackUpperLimit(uint256 buyBackLimit) external onlyOwner() {
872   buyBackUpperLimit = buyBackLimit * 10**15;
873   }
874
875   function isBot(address account) public view returns(bool){
876
```

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

LINE 536

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
535   if (_excluded[i] == account) {
536   _excluded[i] = _excluded[_excluded.length - 1];
537   _tOwned[account] = 0;
538   _isExcluded[account] = false;
539   _excluded.pop();
540
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 15

low SEVERITY

The current pragma Solidity directive is ""^0.8.7"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source File

- FUFTX.sol

Locations

```
14   // SPDX-License-Identifier: NOLICENSE
15   pragma solidity ^0.8.7;
16
17   interface IERC20 {
18   function totalSupply() external view returns (uint256);
19
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 535

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
534   for (uint256 i = 0; i < _excluded.length; i++) {
535   if (_excluded[i] == account) {
536   _excluded[i] = _excluded[_excluded.length - 1];
537   _tOwned[account] = 0;
538   _isExcluded[account] = false;
539
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 536

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
535   if (_excluded[i] == account) {
536   _excluded[i] = _excluded[_excluded.length - 1];
537   _tOwned[account] = 0;
538   _isExcluded[account] = false;
539   _excluded.pop();
540
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 536

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
535   if (_excluded[i] == account) {
536   _excluded[i] = _excluded[_excluded.length - 1];
537   _tOwned[account] = 0;
538   _isExcluded[account] = false;
539   _excluded.pop();
540
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 673

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
672   for (uint256 i = 0; i < _excluded.length; i++) {
673   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
674   rSupply = rSupply-_rOwned[_excluded[i]];
675   tSupply = tSupply-_tOwned[_excluded[i]];
676   }
677
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 673

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
672   for (uint256 i = 0; i < _excluded.length; i++) {
673   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
674   rSupply = rSupply-_rOwned[_excluded[i]];
675   tSupply = tSupply-_tOwned[_excluded[i]];
676   }
677
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 674

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
673   if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
674   rSupply = rSupply-_rOwned[_excluded[i]];
675   tSupply = tSupply-_tOwned[_excluded[i]];
676   }
677   if (rSupply < _rTotal/_tTotal) return (_rTotal, _tTotal);
678
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 675

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
674   rSupply = rSupply-_rOwned[_excluded[i]];
675   tSupply = tSupply-_tOwned[_excluded[i]];
676   }
677   if (rSupply < _rTotal/_tTotal) return (_rTotal, _tTotal);
678   return (rSupply, tSupply);
679
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 770

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
769   address[] memory path = new address[](2);
770   path[0] = router.WETH();
771   path[1] = address(this);
772
773   // make the swap
774
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 771

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
770   path[0] = router.WETH();
771   path[1] = address(this);
772
773   // make the swap
774   router.swapExactETHForTokensSupportingFeeOnTransferTokens{value: amount}(
775
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 820

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
819   address[] memory path = new address[](2);
820   path[0] = address(this);
821   path[1] = router.WETH();
822
823   _approve(address(this), address(router), tokenAmount);
824
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 821

low SEVERITY

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

Source File

- FUFTX.sol

Locations

```
820   path[0] = address(this);
821   path[1] = router.WETH();
822
823   _approve(address(this), address(router), tokenAmount);
824
825
```

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.