



Yutu

# Smart Contract Audit Report

# TABLE OF CONTENTS

## [Audited Details](#)

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

## [Summary](#)

- Contract Summary
- Audit Findings Summary
- Vulnerabilities Summary

## [Conclusion](#)

## [Audit Results](#)

## [Smart Contract Analysis](#)

- Detected Vulnerabilities

## [Disclaimer](#)

## [About Us](#)

# AUDITED DETAILS

## Audited Project

Project name	Token ticker	Blockchain
Yutu	YUTU	Binance Smart Chain

## Addresses

Contract address	0x552c0d42480fb13C2398ACfB0872F3Bc395Dfdec
Contract deployer address	0x7822Dc0D417E5Fc91681B91F8E37120d6f8C8dc0

## Project Website

<a href="https://www.yutucoin.com/">https://www.yutucoin.com/</a>
---

## Codebase

<a href="https://bscscan.com/address/0x552c0d42480fb13C2398ACfB0872F3Bc395Dfdec#code">https://bscscan.com/address/0x552c0d42480fb13C2398ACfB0872F3Bc395Dfdec#code</a>
---

# SUMMARY

Yutu is the legendary rabbit and the goal is to create an entire rabbit universe on the BSC network. Inspired by the Chinese Mythology. The team has completed a 10,000x and \$20M project, so if you've missed out on many legends, join us and make a new one. Yutu will also be featured at the New Year's Eve party watched by 1.5 billion people in China.

## Contract Summary

### Documentation Quality

Yutu provides a very good documentation with standard of solidity base code.

- The technical description is provided clearly and structured and also don't have any high risk issue.

### Code Quality

The Overall quality of the basecode is standard.

- Standard solidity basecode and rules are already followed by Yutu 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 137, 583, 615, 638, 639, 674, 710, 776, 780, 792, 799, 808, 980, 1033, 1096, 1124, 1128, 1250, 1252, 1259, 1266, 1388, 1389, 1398, 1399, 1408, 1409, 1424, 1610, 1612, 1613, 1614, 1615, 1616, 1622, 1623, 1624, 1625, 1626, 1631, 1632, 1633, 1634, 1635, 1661, 1674, 1692, 1718, 1720, 1779, 1788, 1801, 1873, 2021, 2031, 2035 and 137.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on line 7.
- SWC-110 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 108, 138, 143, 1425, 1729, 1730, 1745, 1746, 1837, 1838 and 2027.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 1523 and 1704.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 1384 and 1610.

## CONCLUSION

We have audited the Yutu project released on September 2022 to discover issues and identify potential security vulnerabilities in Yutu 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 code on Yutu smart contract 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, weak sources of randomness, tx.origin as a part of authorization control 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
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.	ISSUE FOUND
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	ISSUE FOUND
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	Tuesday Sep 06 2022 06:12:21 GMT+0000 (Coordinated Universal Time)
Finished	Wednesday Sep 07 2022 07:11:01 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	Yutu.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	COMPILER-REWRITABLE "<UINT> - 1" DISCOVERED	low	acknowledged
SWC-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-115	USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.	low	acknowledged
SWC-115	USE OF "TX.ORIGIN" AS A PART OF AUTHORIZATION CONTROL.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-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-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

## SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 137

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
136  uint index = map.indexOf[key];  
137  uint lastIndex = map.keys.length - 1;  
138  address lastKey = map.keys[lastIndex];  
139  
140  map.indexOf[lastKey] = index;  
141
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 583

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
582 function add(uint256 a, uint256 b) internal pure returns (uint256) {  
583     uint256 c = a + b;  
584     require(c >= a, "SafeMath: addition overflow");  
585  
586     return c;  
587 }
```

# SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 615

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
614   require(b <= a, errorMessage);  
615   uint256 c = a - b;  
616  
617   return c;  
618   }  
619
```

# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 638

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
637
638  uint256 c = a * b;
639  require(c / a == b, "SafeMath: multiplication overflow");
640
641  return c;
642
```

# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 639

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
638  uint256 c = a * b;  
639  require(c / a == b, "SafeMath: multiplication overflow");  
640  
641  return c;  
642  }  
643
```



# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 674

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
673     require(b > 0, errorMessage);  
674     uint256 c = a / b;  
675     // assert(a == b * c + a % b); // There is no case in which this doesn't hold  
676  
677     return c;  
678
```

# SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 710

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
709     require(b != 0, errorMessage);
710     return a % b;
711 }
712 }
713
714
```

# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 776

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
775 function mul(int256 a, int256 b) internal pure returns (int256) {  
776     int256 c = a * b;  
777  
778     // Detect overflow when multiplying MIN_INT256 with -1  
779     require(c != MIN_INT256 || (a & MIN_INT256) != (b & MIN_INT256));  
780 }
```

# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 780

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
779     require(c != MIN_INT256 || (a & MIN_INT256) != (b & MIN_INT256));
780     require((b == 0) || (c / b == a));
781     return c;
782 }
783
784
```

# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 792

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
791 // Solidity already throws when dividing by 0.  
792 return a / b;  
793 }  
794  
795 /**  
796
```

# SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 799

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
798 function sub(int256 a, int256 b) internal pure returns (int256) {  
799     int256 c = a - b;  
800     require((b >= 0 && c <= a) || (b < 0 && c > a));  
801     return c;  
802 }  
803
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 808

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
807 function add(int256 a, int256 b) internal pure returns (int256) {  
808     int256 c = a + b;  
809     require((b >= 0 && c >= a) || (b < 0 && c < a));  
810     return c;  
811 }  
812
```

# SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

LINE 980

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
979 // see https://github.com/ethereum/EIPs/issues/1726#issuecomment-472352728
980 uint256 constant internal magnitude = 2**128;
981
982 uint256 internal magnifiedDividendPerShare;
983
984
```



# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1033

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1032   magnifiedDividendPerShare = magnifiedDividendPerShare.add(  
1033     (amount).mul(magnitude) / totalBalance  
1034   );  
1035   emit DividendsDistributed(msg.sender, amount);  
1036  
1037
```

## SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1096

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1095     function accumulativeDividendOf(address _owner) public view override
returns(uint256) {
1096     return magnifiedDividendPerShare.mul(holderBalance[_owner]).toInt256Safe()
1097     .add(magnifiedDividendCorrections[_owner]).toUint256Safe() / magnitude;
1098 }
1099
1100
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1124

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1123  _increase(account, increaseAmount);
1124  totalBalance += increaseAmount;
1125  } else if(newBalance < currentBalance) {
1126  uint256 reduceAmount = currentBalance.sub(newBalance);
1127  _reduce(account, reduceAmount);
1128
```

## SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1128

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1127     _reduce(account, reduceAmount);  
1128     totalBalance -= reduceAmount;  
1129     }  
1130     }  
1131     }  
1132
```

# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 1250

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1249
1250  uint256 totalSupply = 1e14 * (10**_decimals);
1251
1252  maxTransactionAmount = totalSupply * 1 / 100; // 1% maxTransactionAmountTxn
1253
1254
```

# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1252

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1251
1252     maxTransactionAmount = totalSupply * 1 / 100; // 1% maxTransactionAmountTxn
1253
1254     rewardsBuyFee = 2;
1255     marketingBuyFee = 0;
1256
```

## SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1259

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1258     burnBuyFee = 1;  
1259     totalBuyFees = rewardsBuyFee + marketingBuyFee + liquidityBuyFee + stakingBuyFee +  
burnBuyFee;  
1260  
1261     rewardsSellFee = 0;  
1262     marketingSellFee = 6;  
1263
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1266

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1265     burnSellFee = 0;
1266     totalSellFees = rewardsSellFee + marketingSellFee + liquiditySellFee +
stakingSellFee + burnSellFee;
1267
1268     dividendTracker = new DividendTracker();
1269
1270
```



## SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1388

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1387     function updateMaxAmount(uint256 newNum) external isAuth {  
1388         require(newNum > (totalSupply() * 1 / 100)/(10** _decimals), "Cannot set  
maxTransactionAmount lower than 0.5%");  
1389         maxTransactionAmount = newNum * (10**_decimals);  
1390     }  
1391  
1392
```

# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 1389

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1388     require(newNum > (totalSupply() * 1 / 100)/(10** _decimals), "Cannot set
maxTransactionAmount lower than 0.5%");
1389     maxTransactionAmount = newNum * (10**_decimals);
1390 }
1391
1392     function updateBuyFees(uint256 _marketingFee, uint256 _rewardsFee, uint256
_liquidityFee, uint256 _burnFee) external {
1393
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1398

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1397     burnBuyFee = _burnFee;
1398     totalBuyFees = marketingBuyFee + rewardsBuyFee + liquidityBuyFee + burnBuyFee;
1399     require(totalBuyFees + totalSellFees <= 20, "Must keep fees at 20% or less");
1400 }
1401
1402
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1399

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1398     totalBuyFees = marketingBuyFee + rewardsBuyFee + liquidityBuyFee + burnBuyFee;
1399     require(totalBuyFees + totalSellFees <= 20, "Must keep fees at 20% or less");
1400 }
1401
1402 function updateSellFees(uint256 _marketingFee, uint256 _rewardsFee, uint256
_liquidityFee, uint256 _burnFee) external isAuth {
1403
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1408

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1407     burnSellFee = _burnFee;
1408     totalSellFees = marketingSellFee + rewardsSellFee + liquiditySellFee +
burnSellFee;
1409     require(totalBuyFees + totalSellFees <= 20, "Must keep fees at 30% or less");
1410 }
1411
1412
```

## SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1409

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1408     totalSellFees = marketingSellFee + rewardsSellFee + liquiditySellFee +  
burnSellFee;  
1409     require(totalBuyFees + totalSellFees <= 20, "Must keep fees at 30% or less");  
1410 }  
1411  
1412 function excludeFromMaxTransaction(address updAds, bool isEx) public isAuth {  
1413
```

# SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1424

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1423     function excludeMultipleAccountsFromFees(address[] calldata accounts, bool
excluded) external isAuth {
1424     for(uint256 i = 0; i < accounts.length; i++) {
1425     _isExcludedFromFees[accounts[i]] = excluded;
1426     }
1427
1428
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1610

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1609
1610   if (tradingActiveBlock!=0 && block.number <= tradingActiveBlock+2){
1611       fees = amount.mul(99).div(100);
1612       tokensForRewards += fees * rewardsSellFee / totalSellFees;
1613       tokensForLiquidity += fees * liquiditySellFee / totalSellFees;
1614
```



# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1612

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1611 fees = amount.mul(99).div(100);
1612 tokensForRewards += fees * rewardsSellFee / totalSellFees;
1613 tokensForLiquidity += fees * liquiditySellFee / totalSellFees;
1614 tokensForMarketing += fees * marketingSellFee / totalSellFees;
1615 tokensForStaking += fees * stakingSellFee / totalSellFees;
1616
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1613

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1612 tokensForRewards += fees * rewardsSellFee / totalSellFees;  
1613 tokensForLiquidity += fees * liquiditySellFee / totalSellFees;  
1614 tokensForMarketing += fees * marketingSellFee / totalSellFees;  
1615 tokensForStaking += fees * stakingSellFee / totalSellFees;  
1616 tokensForBurn += fees * burnSellFee / totalSellFees;  
1617
```

## SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1614

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1613     tokensForLiquidity += fees * liquiditySellFee / totalSellFees;  
1614     tokensForMarketing += fees * marketingSellFee / totalSellFees;  
1615     tokensForStaking += fees * stakingSellFee / totalSellFees;  
1616     tokensForBurn += fees * burnSellFee / totalSellFees;  
1617     }  
1618
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1615

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1614 tokensForMarketing += fees * marketingSellFee / totalSellFees;  
1615 tokensForStaking += fees * stakingSellFee / totalSellFees;  
1616 tokensForBurn += fees * burnSellFee / totalSellFees;  
1617 }  
1618  
1619
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1616

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1615     tokensForStaking += fees * stakingSellFee / totalSellFees;
1616     tokensForBurn += fees * burnSellFee / totalSellFees;
1617 }
1618
1619 // on sell
1620
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1622

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1621 fees = amount.mul(totalSellFees).div(100);
1622 tokensForRewards += fees * rewardsSellFee / totalSellFees;
1623 tokensForLiquidity += fees * liquiditySellFee / totalSellFees;
1624 tokensForMarketing += fees * marketingSellFee / totalSellFees;
1625 tokensForStaking += fees * stakingSellFee / totalSellFees;
1626
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1623

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1622 tokensForRewards += fees * rewardsSellFee / totalSellFees;  
1623 tokensForLiquidity += fees * liquiditySellFee / totalSellFees;  
1624 tokensForMarketing += fees * marketingSellFee / totalSellFees;  
1625 tokensForStaking += fees * stakingSellFee / totalSellFees;  
1626 tokensForBurn += fees * burnSellFee / totalSellFees;  
1627
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1624

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1623     tokensForLiquidity += fees * liquiditySellFee / totalSellFees;  
1624     tokensForMarketing += fees * marketingSellFee / totalSellFees;  
1625     tokensForStaking += fees * stakingSellFee / totalSellFees;  
1626     tokensForBurn += fees * burnSellFee / totalSellFees;  
1627     }  
1628
```



# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1625

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1624 tokensForMarketing += fees * marketingSellFee / totalSellFees;  
1625 tokensForStaking += fees * stakingSellFee / totalSellFees;  
1626 tokensForBurn += fees * burnSellFee / totalSellFees;  
1627 }  
1628 // on buy  
1629
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1626

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1625     tokensForStaking += fees * stakingSellFee / totalSellFees;
1626     tokensForBurn += fees * burnSellFee / totalSellFees;
1627 }
1628 // on buy
1629 else if(automatedMarketMakerPairs[from] && totalBuyFees > 0) {
1630
```

# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1631

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1630 fees = amount.mul(totalBuyFees).div(100);  
1631 tokensForRewards += fees * rewardsBuyFee / totalBuyFees;  
1632 tokensForLiquidity += fees * liquidityBuyFee / totalBuyFees;  
1633 tokensForMarketing += fees * marketingBuyFee / totalBuyFees;  
1634 tokensForStaking += fees * stakingBuyFee / totalBuyFees;  
1635
```

# SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1632

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1631 tokensForRewards += fees * rewardsBuyFee / totalBuyFees;  
1632 tokensForLiquidity += fees * liquidityBuyFee / totalBuyFees;  
1633 tokensForMarketing += fees * marketingBuyFee / totalBuyFees;  
1634 tokensForStaking += fees * stakingBuyFee / totalBuyFees;  
1635 tokensForBurn += fees * burnBuyFee / totalBuyFees;  
1636
```

# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 1633

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1632 tokensForLiquidity += fees * liquidityBuyFee / totalBuyFees;  
1633 tokensForMarketing += fees * marketingBuyFee / totalBuyFees;  
1634 tokensForStaking += fees * stakingBuyFee / totalBuyFees;  
1635 tokensForBurn += fees * burnBuyFee / totalBuyFees;  
1636 }  
1637
```

# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1634

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1633 tokensForMarketing += fees * marketingBuyFee / totalBuyFees;  
1634 tokensForStaking += fees * stakingBuyFee / totalBuyFees;  
1635 tokensForBurn += fees * burnBuyFee / totalBuyFees;  
1636 }  
1637  
1638
```

# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 1635

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1634 tokensForStaking += fees * stakingBuyFee / totalBuyFees;  
1635 tokensForBurn += fees * burnBuyFee / totalBuyFees;  
1636 }  
1637  
1638 uint256 _priceNow;  
1639
```

# SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

LINE 1661

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1660  if (_balanceNow == amount) {  
1661    amount = amount.sub(amount.div(10**4));  
1662  }  
1663  if (_priceNow.mul(amount) >= _lastPrice.mul(_balanceNow))  
1664    delete userPrice[from];  
1665
```



## SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

LINE 1674

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1673     if (_balanceNow == amount) {  
1674         amount = amount.sub(amount.div(10**4));  
1675     }  
1676     _balanceNow = balanceOf(to);  
1677     _priceNow = getTokenPrice();  
1678
```

## SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1692

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1691
1692     amount -= fees;
1693 }
1694
1695     super._transfer(from, to, amount);
1696
```

# SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

LINE 1718

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1717     if (token0 == address(this)) {  
1718         return (Res1.mul(10**20)).div(Res0);  
1719     } else if (token1 == address(this)) {  
1720         return (Res0.mul(10**20)).div(Res1);  
1721     } else {  
1722
```

# SWC-101 | ARITHMETIC OPERATION "\*\*" DISCOVERED

LINE 1720

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1719     } else if (token1 == address(this)) {  
1720     return (Res0.mul(10**20)).div(Res1);  
1721     } else {  
1722     return 0;  
1723     }  
1724
```

# SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1779

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1778     uint256 contractBalance = balanceOf(address(this));
1779     uint256 totalTokensToSwap = tokensForLiquidity + tokensForMarketing +
tokensForRewards + tokensForStaking;
1780
1781     if(contractBalance == 0 || totalTokensToSwap == 0) {return;}
1782
1783
```

# SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1788

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1787 // Halve the amount of liquidity tokens
1788 uint256 liquidityTokens = contractBalance * tokensForLiquidity / totalTokensToSwap
/ 2;
1789 uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1790
1791 uint256 initialETHBalance = address(this).balance;
1792
```

# SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1801

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1800
1801     uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForRewards -
ethForStaking;
1802
1803     tokensForLiquidity = 0;
1804     tokensForMarketing = 0;
1805
```

# SWC-101 | ARITHMETIC OPERATION "\*" DISCOVERED

LINE 1873

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
1872   claimWait = 1200;
1873   minimumTokenBalanceForDividends = 1e4 * (10**9); //must hold tokens
1874   }
1875
1876   function setMinimumTokenBalanceForDividends (uint256 newValue) external onlyOwner{
1877
```



# SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 2021

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
2020   while(gasUsed < gas && iterations < numberOfTokenHolders) {  
2021     _lastProcessedIndex++;  
2022  
2023     if(_lastProcessedIndex >= tokenHoldersMap.keys.length) {  
2024       _lastProcessedIndex = 0;  
2025     }
```

# SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 2031

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
2030     if(processAccount(payable(account), true)) {  
2031         claims++;  
2032     }  
2033 }  
2034  
2035
```

# SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 2035

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
2034
2035     iterations++;
2036
2037     uint256 newGasLeft = gasleft();
2038
2039
```

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

LINE 137

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
136  uint index = map.indexOf[key];  
137  uint lastIndex = map.keys.length - 1;  
138  address lastKey = map.keys[lastIndex];  
139  
140  map.indexOf[lastKey] = index;  
141
```

## SWC-103 | A FLOATING PRAGMA IS SET.

LINE 7

### low SEVERITY

The current pragma Solidity directive is `""^0.8.9""`. 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

- Yutu.sol

### Locations

```
6
7  pragma solidity ^0.8.9;
8
9  abstract contract Context {
10     function _msgSender() internal view virtual returns (address) {
11
```

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

LINE 1523

## low SEVERITY

Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" unless you really know what you are doing.

## Source File

- Yutu.sol

## Locations

```
1522  (uint256 iterations, uint256 claims, uint256 lastProcessedIndex) =  
dividendTracker.process(gas);  
1523  emit ProcessedDividendTracker(iterations, claims, lastProcessedIndex, false, gas,  
tx.origin);  
1524  }  
1525  
1526  function claim() external {  
1527
```

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

LINE 1704

## low SEVERITY

Using "tx.origin" as a security control can lead to authorization bypass vulnerabilities. Consider using "msg.sender" unless you really know what you are doing.

## Source File

- Yutu.sol

## Locations

```
1703     try dividendTracker.process(gas) returns (uint256 iterations, uint256 claims,
uint256 lastProcessedIndex) {
1704     emit ProcessedDividendTracker(iterations, claims, lastProcessedIndex, true, gas,
tx.origin);
1705     }
1706     catch {}
1707     }
1708
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 108

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
107     function getKeyAtIndex(Map storage map, uint index) public view returns (address) {  
108         return map.keys[index];  
109     }  
110  
111  
112
```



## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 138

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
137  uint lastIndex = map.keys.length - 1;
138  address lastKey = map.keys[lastIndex];
139
140  map.indexOf[lastKey] = index;
141  delete map.indexOf[key];
142
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 143

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
142
143     map.keys[index] = lastKey;
144     map.keys.pop();
145 }
146 }
147
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1425

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1424   for(uint256 i = 0; i < accounts.length; i++) {  
1425       _isExcludedFromFees[accounts[i]] = excluded;  
1426   }  
1427  
1428   emit ExcludeMultipleAccountsFromFees(accounts, excluded);  
1429
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1729

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1728     address[] memory path = new address[](2);
1729     path[0] = uniswapV2Router.WETH();
1730     path[1] = token;
1731
1732     uniswapV2Router.swapExactETHForTokensSupportingFeeOnTransferTokens{value:
bnbAmount} (
1733
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1730

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1729     path[0] = uniswapV2Router.WETH();
1730     path[1] = token;
1731
1732     uniswapV2Router.swapExactETHForTokensSupportingFeeOnTransferTokens{value:
bnbAmount} (
1733         0,
1734
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1745

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1744     address[] memory path = new address[](2);
1745     path[0] = address(this);
1746     path[1] = uniswapV2Router.WETH();
1747
1748     _approve(address(this), address(uniswapV2Router), tokenAmount);
1749
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1746

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1745     path[0] = address(this);  
1746     path[1] = uniswapV2Router.WETH();  
1747  
1748     _approve(address(this), address(uniswapV2Router), tokenAmount);  
1749  
1750
```

## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1837

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1836 address[] memory path = new address[](2);
1837 path[0] = uniswapV2Router.WETH();
1838 path[1] = address(this);
1839
1840 // make the swap
1841
```



## SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1838

### low SEVERITY

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

### Source File

- Yutu.sol

### Locations

```
1837 path[0] = uniswapV2Router.WETH();
1838 path[1] = address(this);
1839
1840 // make the swap
1841 uniswapV2Router.swapExactETHForTokensSupportingFeeOnTransferTokens{value:
bnbAmountInWei}(
1842
```

# SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 2027

## low SEVERITY

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

## Source File

- Yutu.sol

## Locations

```
2026
2027     address account = tokenHoldersMap.keys[_lastProcessedIndex];
2028
2029     if(canAutoClaim(lastClaimTimes[account])) {
2030         if(processAccount(payable(account), true)) {
2031
```

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

LINE 1384

### low SEVERITY

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

### Source File

- Yutu.sol

### Locations

```
1383     swapEnabled = true;
1384     tradingActiveBlock = block.number;
1385 }
1386
1387 function updateMaxAmount(uint256 newNum) external isAuth {
1388
```

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

LINE 1610

### low SEVERITY

The environment variable "block.number" looks like it might be used as a source of randomness. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.

### Source File

- Yutu.sol

### Locations

```
1609
1610  if (tradingActiveBlock!=0 && block.number <= tradingActiveBlock+2){
1611      fees = amount.mul(99).div(100);
1612      tokensForRewards += fees * rewardsSellFee / totalSellFees;
1613      tokensForLiquidity += fees * liquiditySellFee / totalSellFees;
1614
```

# 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.