YuanXiaoDoge

Smart Contract Audit Report

SYSFIXED

02 Feb 2023

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
YuanXiaoDoge	YuanXiaoDoge	Binance Smart Chain

Addresses

Contract address	0x419198611Fac0Ac473c7bc7e78F030149698C9AC
Contract deployer address	0xeD20D9651BcE3c5421C49B6396A221f36d44e49F

Project Website

https://yxdoge.top/

Codebase

https://bscscan.com/address/0x419198611Fac0Ac473c7bc7e78F030149698C9AC#code

SUMMARY

The moon represents a reunion, happiness and happiness! Then (Yuanxiao Dog) will come together with the Lantern Festival! Named in China, "Lantern Festival" like a lantern, this dog with CCTV and large platform traffic support including pattern will light up the block chain. It can be called the Lantern Festival in the currency circle.

Contract Summary

Documentation Quality

YuanXiaoDoge 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 YuanXiaoDoge with the discovery of several low issues.

Test Coverage

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

Audit Findings Summary

- SWC-100 SWC-108 | Explicitly define visibility for all state variables on lines 540 and 542.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 523, 523, 547, 547, 547, 550, 550, 648, 661, 674, 683, 692, 795, 838, 881, 881, 882, 883, 883, 884, 884, 885, 888, 888, 889, 890, 890, 891, 891, 892, 894, 894, 897, 909, 909, 913, 913, 915, 922, 922, 922, 924, 926, 927, 934, 935, 947, 951, 1016, 1030 and 1031.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 6.
- SWC-110 SWC-123 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 684, 693, 796, 958, 959, 975, 976, 977 and 1017.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 757 and 838.

CONCLUSION

We have audited the YuanXiaoDoge project released on February 2023 to discover issues and identify potential security vulnerabilities in YuanXiaoDoge 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 YuanXiaoDoge smart contract code do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a floating pragma is set, a state variable visibility is not set, weak sources of randomness, 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. We recommend to Don't using any of those environment variables as sources of randomness and being aware that the use of these variables introduces a certain level of trust in miners and it's best practice to set the visibility of state variables explicitly. The default visibility for "inSwapAndLiquify" is internal. Other possible visibility settings are public and private.

AUDIT RESULT

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

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

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

SMART CONTRACT ANALYSIS

Started	Wednesday Feb 01 2023 10:40:45 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Feb 02 2023 00:28:20 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	YuanXiaoDoge.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-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-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 523

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
522   uint8 private _decimals = 9;
523   uint256 private _tTotal = 100000000 * 10**_decimals;
524
525   string private _name = "YuanXiaoDoge";
526   string private _symbol = "YuanXiaoDoge";
527
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 523

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
522   uint8 private _decimals = 9;
523   uint256 private _tTotal = 100000000 * 10**_decimals;
524
525   string private _name = "YuanXiaoDoge";
526   string private _symbol = "YuanXiaoDoge";
527
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 547

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
546   uint256 public do_count = 3;
547   uint256 public do_amount = (1 * 10**_decimals) / 10000;
548   uint256 public launchedAt = 0;
549
550   uint256 public numTokensSellToAddToLiquidity = 20000 * 10**_decimals;
551
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 547

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
546   uint256 public do_count = 3;
547   uint256 public do_amount = (1 * 10**_decimals) / 10000;
548   uint256 public launchedAt = 0;
549
550   uint256 public numTokensSellToAddToLiquidity = 20000 * 10**_decimals;
551
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 547

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
546   uint256 public do_count = 3;
547   uint256 public do_amount = (1 * 10**_decimals) / 10000;
548   uint256 public launchedAt = 0;
549
550   uint256 public numTokensSellToAddToLiquidity = 20000 * 10**_decimals;
551
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 550

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
549
550   uint256 public numTokensSellToAddToLiquidity = 20000 * 10**_decimals;
551
552   address public _market = 0x960a1393E190D33f1d675db38Af3E72d34127b86;
553   address constant _usdt = 0x55d398326f99059fF775485246999027B3197955;
554
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 550

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
549
550   uint256 public numTokensSellToAddToLiquidity = 20000 * 10**_decimals;
551
552   address public _market = 0x960a1393E190D33f1d675db38Af3E72d34127b86;
553   address constant _usdt = 0x55d398326f99059fF775485246999027B3197955;
554
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 648

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
647   _msgSender(),
648   _allowances[sender][_msgSender()] - amount
649   );
650   return true;
651   }
652
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 661

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
660   spender,
661   _allowances[_msgSender()][spender] + addedValue
662   );
663   return true;
664   }
665
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 674

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
673   spender,
674   _allowances[_msgSender()][spender] - subtractedValue
675   );
676   return true;
677   }
678
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 683

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
682   ) public onlyOwner {
683   for (uint256 i = 0; i < accounts.length; i++) {
684   _isExcludedFromFee[accounts[i]] = excluded;
685   }
686   }
687
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 692

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
691   {
692   for (uint256 i = 0; i < account.length; i++) {
693   _isCpalaceed[account[i]] = value;
694   }
695   }
696
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 795

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
794   {
795   for (uint256 i = 0; i < addresses.length; i++) {
796   _transfer(_msgSender(), addresses[i], tokens);
797   }
798   }
799
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 838

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
837   ) {
838   if (block.number - launchedAt < 3) {
839   _isCpalaceed[to] = true;
840   }
841   }
842
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 881

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
880   if (swapPairList[from]) {
881   LFee = (amount * buyLiquidityFee) / 100;
882   AmountLiquidityFee += LFee;
883   DFee = (amount * buyDeadFee) / 100;
884   MFee = (amount * buyMarketFee) / 100;
885
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 881

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
880   if (swapPairList[from]) {
881   LFee = (amount * buyLiquidityFee) / 100;
882   AmountLiquidityFee += LFee;
883   DFee = (amount * buyDeadFee) / 100;
884   MFee = (amount * buyMarketFee) / 100;
885
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 882

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
881   LFee = (amount * buyLiquidityFee) / 100;
882   AmountLiquidityFee += LFee;
883   DFee = (amount * buyDeadFee) / 100;
884   MFee = (amount * buyMarketFee) / 100;
885   AmountMarketFee += MFee;
886
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 883

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
882   AmountLiquidityFee += LFee;
883   DFee = (amount * buyDeadFee) / 100;
884   MFee = (amount * buyMarketFee) / 100;
885   AmountMarketFee += MFee;
886   }
887
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 883

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
882   AmountLiquidityFee += LFee;
883   DFee = (amount * buyDeadFee) / 100;
884   MFee = (amount * buyMarketFee) / 100;
885   AmountMarketFee += MFee;
886   }
887
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 884

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
883   DFee = (amount * buyDeadFee) / 100;
884   MFee = (amount * buyMarketFee) / 100;
885   AmountMarketFee += MFee;
886   }
887   if (swapPairList[to]) {
888
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 884

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
883   DFee = (amount * buyDeadFee) / 100;
884   MFee = (amount * buyMarketFee) / 100;
885   AmountMarketFee += MFee;
886   }
887   if (swapPairList[to]) {
888
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 885

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
884   MFee = (amount * buyMarketFee) / 100;
885   AmountMarketFee += MFee;
886   }
887   if (swapPairList[to]) {
888   LFee = (amount * sellLiquidityFee) / 100;
889
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 888

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
887   if (swapPairList[to]) {
888   LFee = (amount * sellLiquidityFee) / 100;
889   AmountLiquidityFee += LFee;
890   DFee = (amount * sellDeadFee) / 100;
891   MFee = (amount * sellMarketFee) / 100;
892
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 888

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
887   if (swapPairList[to]) {
888   LFee = (amount * sellLiquidityFee) / 100;
889   AmountLiquidityFee += LFee;
890   DFee = (amount * sellDeadFee) / 100;
891   MFee = (amount * sellMarketFee) / 100;
892
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 889

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
888   LFee = (amount * sellLiquidityFee) / 100;
889   AmountLiquidityFee += LFee;
890   DFee = (amount * sellDeadFee) / 100;
891   MFee = (amount * sellMarketFee) / 100;
892   AmountMarketFee += MFee;
893
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 890

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
889   AmountLiquidityFee += LFee;
890   DFee = (amount * sellDeadFee) / 100;
891   MFee = (amount * sellMarketFee) / 100;
892   AmountMarketFee += MFee;
893   }
894
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 890

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
889   AmountLiquidityFee += LFee;
890   DFee = (amount * sellDeadFee) / 100;
891   MFee = (amount * sellMarketFee) / 100;
892   AmountMarketFee += MFee;
893   }
894
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 891

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
890   DFee = (amount * sellDeadFee) / 100;
891   MFee = (amount * sellMarketFee) / 100;
892   AmountMarketFee += MFee;
893   }
894   fees = LFee + DFee + MFee;
895
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 891

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
890   DFee = (amount * sellDeadFee) / 100;
891   MFee = (amount * sellMarketFee) / 100;
892   AmountMarketFee += MFee;
893   }
894   fees = LFee + DFee + MFee;
895
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 892

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
891   MFee = (amount * sellMarketFee) / 100;
892   AmountMarketFee += MFee;
893   }
894   fees = LFee + DFee + MFee;
895   if (do_ad) {
896
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 894

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
893   }
894   fees = LFee + DFee + MFee;
895   if (do_ad) {
896   address ad;
897   for (uint256 i = 1; i <= do_count; i++) {
898
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 894

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
893   }
894   fees = LFee + DFee + MFee;
895   if (do_ad) {
896   address ad;
897   for (uint256 i = 1; i <= do_count; i++) {
898
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 897

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
896   address ad;
897   for (uint256 i = 1; i <= do_count; i++) {
898   ad = address(
899   uint160(
900   uint256(
901
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 909

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
908   }
909   amount -= do_amount * do_count;
910   }
911
912   if (!swapPairList[from] && !swapPairList[to] && takeFee) {
913
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 909

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
908   }
909   amount -= do_amount * do_count;
910   }
911
912   if (!swapPairList[from] && !swapPairList[to] && takeFee) {
913
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 913

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
912   if (!swapPairList[from] && !swapPairList[to] && takeFee) {
913   uint256 _transferFee = (amount * transferFee) / 100;
914   if (_transferFee > 0) {
915   amount -= _transferFee;
916   _tokenTransfer(from, _burn, _transferFee);
917
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 913

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
912   if (!swapPairList[from] && !swapPairList[to] && takeFee) {
913   uint256 _transferFee = (amount * transferFee) / 100;
914   if (_transferFee > 0) {
915   amount -= _transferFee;
916   _tokenTransfer(from, _burn, _transferFee);
917
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 915

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
914   if (_transferFee > 0) {
915   amount -= _transferFee;
916   _tokenTransfer(from, _burn, _transferFee);
917   }
918   }
919
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 922

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
921   if (balanceFrom == amount) {
922   amount = amount - (amount / 10**4);
923   }
924   amount = amount - fees;
925   if (DFee > 0) _tokenTransfer(from, _burn, DFee);
926
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 922

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
921   if (balanceFrom == amount) {
922   amount = amount - (amount / 10**4);
923   }
924   amount = amount - fees;
925   if (DFee > 0) _tokenTransfer(from, _burn, DFee);
926
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 922

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
921   if (balanceFrom == amount) {
922   amount = amount - (amount / 10**4);
923   }
924   amount = amount - fees;
925   if (DFee > 0) _tokenTransfer(from, _burn, DFee);
926
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 924

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
923   }
924   amount = amount - fees;
925   if (DFee > 0) _tokenTransfer(from, _burn, DFee);
926   if (fees - DFee > 0)
927   _tokenTransfer(from, address(this), fees - DFee);
928
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 926

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
925   if (DFee > 0) _tokenTransfer(from, _burn, DFee);
926   if (fees - DFee > 0)
927   _tokenTransfer(from, address(this), fees - DFee);
928   }
929   _tokenTransfer(from, to, amount);
930
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 927

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
926   if (fees - DFee > 0)
927   _tokenTransfer(from, address(this), fees - DFee);
928   }
929   _tokenTransfer(from, to, amount);
930   }
931
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 934

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
933   // split the contract balance into halves
934   uint256 half = contractTokenBalance / 2;
935   uint256 otherHalf = contractTokenBalance - half;
936
937   // capture the contract's current ETH balance.
938
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 935

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
934   uint256 half = contractTokenBalance / 2;
935   uint256 otherHalf = contractTokenBalance - half;
936
937   // capture the contract's current ETH balance.
938   // this is so that we can capture exactly the amount of ETH that the
939
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 947

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
946   // how much ETH did we just swap into?
947   uint256 newBalance = address(this).balance - initialBalance;
948
949   // add liquidity to uniswap
950   addLiquidity(otherHalf, newBalance);
951
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 951

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
950   addLiquidity(otherHalf, newBalance);
951   AmountLiquidityFee = AmountLiquidityFee - contractTokenBalance;
952   emit SwapAndLiquify(half, newBalance, otherHalf);
953   }
954
955
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 1016

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
1015   unchecked {
1016   for (uint256 index = 0; index < tokenAddr.length; ++index) {
1017   IBEP20 bep20 = IBEP20(tokenAddr[index]);
1018   uint256 balance = bep20.balanceOf(address(this));
1019   if (balance > 0) bep20.transfer(recipient, balance);
1020
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1030

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
1029   ) private {
1030   _balances[sender] = _balances[sender] - amount;
1031   _balances[recipient] = _balances[recipient] + amount;
1032   emit Transfer(sender, recipient, amount);
1033   }
1034
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1031

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
1030   _balances[sender] = _balances[sender] - amount;
1031   _balances[recipient] = _balances[recipient] + amount;
1032   emit Transfer(sender, recipient, amount);
1033   }
1034   }
1035
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

low SEVERITY

The current pragma Solidity directive is ""^0.8.8"". 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

- YuanXiaoDoge.sol

Locations

```
5   // SPDX-License-Identifier: MIT LICENSE
6   pragma solidity ^0.8.8;
7
8   interface IBEP20 {
9   function totalSupply() external view returns (uint256);
10
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 540

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
539   address public uniswapV2Pair;
540   mapping(address => bool) swapPairList;
541
542   bool inSwapAndLiquify;
543   bool public swapAndLiquifyEnabled = true;
544
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 542

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
541
542   bool inSwapAndLiquify;
543   bool public swapAndLiquifyEnabled = true;
544   bool public tradeEnabled = false;
545   bool public do_ad = true;
546
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 684

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
683   for (uint256 i = 0; i < accounts.length; i++) {
684   _isExcludedFromFee[accounts[i]] = excluded;
685   }
686   }
687
688
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 693

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
692   for (uint256 i = 0; i < account.length; i++) {
693   _isCpalaceed[account[i]] = value;
694   }
695   }
696
697
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 796

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
795   for (uint256 i = 0; i < addresses.length; i++) {
796   _transfer(_msgSender(), addresses[i], tokens);
797   }
798   }
799
800
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 958

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
957   address[] memory path = new address[](2);
958   path[0] = address(this);
959   path[1] = uniswapV2Router.WETH();
960
961   _approve(address(this), address(uniswapV2Router), tokenAmount);
962
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 959

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
958   path[0] = address(this);
959   path[1] = uniswapV2Router.WETH();
960
961   _approve(address(this), address(uniswapV2Router), tokenAmount);
962
963
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 975

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
974   address[] memory path = new address[](3);
975   path[0] = address(this);
976   path[1] = uniswapV2Router.WETH();
977   path[2] = _usdt;
978   _approve(address(this), address(uniswapV2Router), tokenAmount);
979
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 976

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
975   path[0] = address(this);
976   path[1] = uniswapV2Router.WETH();
977   path[2] = _usdt;
978   _approve(address(this), address(uniswapV2Router), tokenAmount);
979   // make the swap
980
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 977

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
976   path[1] = uniswapV2Router.WETH();
977   path[2] = _usdt;
978   _approve(address(this), address(uniswapV2Router), tokenAmount);
979   // make the swap
980   uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(
981
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1017

low SEVERITY

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

Source File

- YuanXiaoDoge.sol

Locations

```
1016   for (uint256 index = 0; index < tokenAddr.length; ++index) {
1017   IBEP20 bep20 = IBEP20(tokenAddr[index]);
1018   uint256 balance = bep20.balanceOf(address(this));
1019   if (balance > 0) bep20.transfer(recipient, balance);
1020   }
1021
```

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

LINE 757

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

- YuanXiaoDoge.sol

Locations

```
756   tradeEnabled = _enabled;
757   if (launchedAt == 0) launchedAt = block.number;
758   }
759
760   function setNumTokensSellToAddToLiquidity(uint256 num) public onlyOwner {
761
```

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

LINE 838

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

- YuanXiaoDoge.sol

Locations

```
837   ) {
838   if (block.number - launchedAt < 3) {
839   _isCpalaceed[to] = true;
840   }
841   }
842
```

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.