Custodiy (V3)

Smart Contract Audit Report

SYSFIXED

09 Feb 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Custodiy (V3)	CTY	Binance Smart Chain

Addresses

Contract address	0xba08da6b46e3dd153dd8b66a6e4cfd37a6359559
Contract deployer address	0x0D2CA37916B670685553416D220378485171dca3

Project Website

https://www.custodiy.com/

Codebase

https://bscscan.com/address/0xba08da6b46e3dd153dd8b66a6e4cfd37a6359559#code

SUMMARY

The project was born in 2019. A team of Italian programmers supported by the Canadian company AMCO IT for code programming in the blockchain sector decided to conceive the project and implement the first drafts of the service. At the same time, the team began to expand, and the first collaborations with centralized banking institutions started to materialize. The same growth and Marketing developed within the project allow Custodiy to establish its internal community linked together by the vision of the development of the project and of the functions of the same web app already existing at CUSTODIY.COM. The token related to the Custodiy project is CTY $, a ticket with the availability of 1,000,000 units.

Contract Summary

Documentation Quality

Custodiy (V3) 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 Custodiy (V3) 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 204, 226, 251, 280, 281, 410, 410, 411, 411, 412, 412, 413, 413, 443, 443, 473, 483, 494, 512, 523, 534, 552, 552, 559, 559, 566, 566, 573, 573, 580, 584, 584, 604, 605, 605, 607, 613, 614, 614, 615, 622, 622, 623, 623, 675, 675, 684, 684, 693, 693, 702, 702, 729, 742, 742, 743, 743, 744 and 744.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 17.
- 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 637, 638 and 730.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 512 and 714.

CONCLUSION

We have audited the Custodiy (V3) project released on February 2023 to discover issues and identify potential security vulnerabilities in Custodiy (V3) 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 satisfactory results with low-risk issues.

The issues in the Custodiy (V3) 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, the potential use of "block.number" as a source of randomness, out-of-bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value.

AUDIT RESULT

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

DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS
Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	PASS
Block values as a proxy for time	SWC-116	Block numbers should not be used for time calculations.	PASS
Signature Unique ID	SWC-117 SWC-121 SWC-122	Signed messages should always have a unique id. A transaction hash should not be used as a unique id.	PASS
Incorrect Constructor Name	SWC-118	Constructors are special functions that are called only once during the contract creation.	PASS
Shadowing State Variable	SWC-119	State variables should not be shadowed.	PASS
Weak Sources of Randomness	SWC-120	Random values should never be generated from Chain Attributes or be predictable.	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 08 2023 03:37:11 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Feb 09 2023 13:22:42 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	CustodiyV3.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-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-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 204

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
203   require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
204   _approve(sender, _msgSender(), currentAllowance - amount);
205
206   return true;
207   }
208
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 226

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
225   {
226   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
227   return true;
228   }
229
230
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 251

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
250   require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
251   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
252
253   return true;
254   }
255
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 280

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
279   require(senderBalance >= amount, "ERC20: transfer amount exceeds balance");
280   _balances[sender] = senderBalance - amount;
281   _balances[recipient] += amount;
282
283   emit Transfer(sender, recipient, amount);
284
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 281

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
280   _balances[sender] = senderBalance - amount;
281   _balances[recipient] += amount;
282
283   emit Transfer(sender, recipient, amount);
284   }
285
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 410

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
409
410   uint256 public tokenLiquidityThreshold = 1_000 * 10**18;
411   uint256 public maxBuyLimit = 10_000 * 10**18;
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 410

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
409
410   uint256 public tokenLiquidityThreshold = 1_000 * 10**18;
411   uint256 public maxBuyLimit = 10_000 * 10**18;
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 411

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
410   uint256 public tokenLiquidityThreshold = 1_000 * 10**18;
411   uint256 public maxBuyLimit = 10_000 * 10**18;
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
415
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 411

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
410   uint256 public tokenLiquidityThreshold = 1_000 * 10**18;
411   uint256 public maxBuyLimit = 10_000 * 10**18;
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
415
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 412

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
411   uint256 public maxBuyLimit = 10_000 * 10**18;
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
415   uint256 public genesis_block;
416
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 412

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
411   uint256 public maxBuyLimit = 10_000 * 10**18;
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
415   uint256 public genesis_block;
416
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 413

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
415   uint256 public genesis_block;
416
417
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 413

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
412   uint256 public maxSellLimit = 10_000 * 10**18;
413   uint256 public maxWalletLimit = 20_000 * 10**18;
414
415   uint256 public genesis_block;
416
417
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 443

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
442   constructor() ERC20("Custodiy (V3)", "CTY") {
443   _tokengeneration(msg.sender, 1_000_000 * 10**decimals());
444   exemptFee[msg.sender] = true;
445
446   // IRouter _router = IRouter(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D); //
UNISWAP V2
447
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 443

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
442   constructor() ERC20("Custodiy (V3)", "CTY") {
443   _tokengeneration(msg.sender, 1_000_000 * 10**decimals());
444   exemptFee[msg.sender] = true;
445
446   // IRouter _router = IRouter(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D); //
UNISWAP V2
447
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 473

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
472   require(currentAllowance >= amount, "ERC20: transfer amount exceeds allowance");
473   _approve(sender, _msgSender(), currentAllowance - amount);
474
475   return true;
476   }
477
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 483

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
482   {
483   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
484   return true;
485   }
486
487
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 494

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
493   require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below
zero");
494   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
495
496   return true;
497   }
498
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 512

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
511
512   if(block.number < genesis_block + 50 && sender == pair) {
513   nonCustodial[recipient] = true;
514   }
515
516
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 523

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
522   require(
523   balanceOf(recipient) + amount <= maxWalletLimit,
524   "You are exceeding maxWalletLimit"
525   );
526   }
527
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 534

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
533   require(
534   balanceOf(recipient) + amount <= maxWalletLimit,
535   "You are exceeding maxWalletLimit"
536   );
537   }
538
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 552

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
551   feeswap =
552   sellTaxes.liquidity +
553   sellTaxes.marketing +
554   sellTaxes.developer;
555   feesum = feeswap;
556
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 552

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
551   feeswap =
552   sellTaxes.liquidity +
553   sellTaxes.marketing +
554   sellTaxes.developer;
555   feesum = feeswap;
556
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 559

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
558   feeswap =
559   taxes.liquidity +
560   taxes.marketing +
561   taxes.developer ;
562   feesum = feeswap;
563
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 559

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
558   feeswap =
559   taxes.liquidity +
560   taxes.marketing +
561   taxes.developer ;
562   feesum = feeswap;
563
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 566

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
565   feeswap =
566   transferTaxes.liquidity +
567   transferTaxes.marketing +
568   transferTaxes.developer ;
569   feesum = feeswap;
570
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 566

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
565   feeswap =
566   transferTaxes.liquidity +
567   transferTaxes.marketing +
568   transferTaxes.developer ;
569   feesum = feeswap;
570
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 573

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
572
573   fee = (amount * feesum) / 100;
574
575   //send fees if threshold has been reached
576   //don't do this on buys, breaks swap
577
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 573

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
572
573   fee = (amount * feesum) / 100;
574
575   //send fees if threshold has been reached
576   //don't do this on buys, breaks swap
577
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 580

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
579   //rest to recipient
580   super._transfer(sender, recipient, amount - fee);
581   if (fee > 0) {
582   //send the fee to the contract
583   if (feeswap > 0) {
584
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 584

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
583   if (feeswap > 0) {
584   uint256 feeAmount = (amount * feeswap) / 100;
585   super._transfer(sender, address(this), feeAmount);
586   }
587
588
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 584

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
583   if (feeswap > 0) {
584   uint256 feeAmount = (amount * feeswap) / 100;
585   super._transfer(sender, address(this), feeAmount);
586   }
587
588
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 604

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
603   // Split the contract balance into halves
604   uint256 denominator = feeswap * 2;
605   uint256 tokensToAddLiquidityWith = (contractBalance * swapTaxes.liquidity) /
606   denominator;
607   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
608
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 605

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
604   uint256 denominator = feeswap * 2;
605   uint256 tokensToAddLiquidityWith = (contractBalance * swapTaxes.liquidity) /
606   denominator;
607   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
608
609
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 605

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
604   uint256 denominator = feeswap * 2;
605   uint256 tokensToAddLiquidityWith = (contractBalance * swapTaxes.liquidity) /
606   denominator;
607   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
608
609
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 607

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
606   denominator;
607   uint256 toSwap = contractBalance - tokensToAddLiquidityWith;
608
609   uint256 initialBalance = address(this).balance;
610
611
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 613

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
612
613   uint256 deltaBalance = address(this).balance - initialBalance;
614   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
615   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
616
617
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 614

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
613   uint256 deltaBalance = address(this).balance - initialBalance;
614   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
615   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
616
617   if (ethToAddLiquidityWith > 0) {
618
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 614

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
613   uint256 deltaBalance = address(this).balance - initialBalance;
614   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
615   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
616
617   if (ethToAddLiquidityWith > 0) {
618
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 615

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
614   uint256 unitBalance = deltaBalance / (denominator - swapTaxes.liquidity);
615   uint256 ethToAddLiquidityWith = unitBalance * swapTaxes.liquidity;
616
617   if (ethToAddLiquidityWith > 0) {
618   // Add liquidity to pancake
619
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 622

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
621
622   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
623   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
624   if (marketingAmt > 0) {
625   payable(marketingWallet).sendValue(marketingAmt);
626
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 622

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
621
622   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
623   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
624   if (marketingAmt > 0) {
625   payable(marketingWallet).sendValue(marketingAmt);
626
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 623

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
622   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
623   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
624   if (marketingAmt > 0) {
625   payable(marketingWallet).sendValue(marketingAmt);
626   }
627
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 623

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
622   uint256 marketingAmt = unitBalance * 2 * swapTaxes.marketing;
623   uint256 developerAmt = unitBalance * 2 * swapTaxes.developer;
624   if (marketingAmt > 0) {
625   payable(marketingWallet).sendValue(marketingAmt);
626   }
627
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 675

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
674   require(new_amount <= 20_000 && new_amount > 0, "Swap threshold amount should be
lower or euqal to 1% of tokens");
675   tokenLiquidityThreshold = new_amount * 10**decimals();
676   }
677
678   function SetBuyTaxes(
679
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 675

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
674   require(new_amount <= 20_000 && new_amount > 0, "Swap threshold amount should be
lower or euqal to 1% of tokens");
675   tokenLiquidityThreshold = new_amount * 10**decimals();
676   }
677
678   function SetBuyTaxes(
679
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 684

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
683   taxes = Taxes(_marketing, _liquidity,  _developer);
684   require((_marketing + _liquidity +  _developer) <= 15, "Must keep fees at 15% or
less");
685   }
686
687   function SetSellTaxes(
688
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 684

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
683   taxes = Taxes(_marketing, _liquidity,  _developer);
684   require((_marketing + _liquidity +  _developer) <= 15, "Must keep fees at 15% or
less");
685   }
686
687   function SetSellTaxes(
688
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 693

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
692   sellTaxes = Taxes(_marketing, _liquidity,  _developer);
693   require((_marketing + _liquidity + _developer) <= 99, "Must keep fees at 99% or
less");
694   }
695
696   function SetTransferTaxes(
697
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 693

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
692   sellTaxes = Taxes(_marketing, _liquidity,  _developer);
693   require((_marketing + _liquidity + _developer) <= 99, "Must keep fees at 99% or
less");
694   }
695
696   function SetTransferTaxes(
697
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 702

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
701   transferTaxes = Taxes(_marketing, _liquidity,  _developer);
702   require((_marketing + _liquidity + _developer) <= 99, "Must keep fees at 99% or
less");
703   }
704
705   function updateRouterAndPair(address newRouter, address newPair) external onlyOwner
{
706
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 702

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
701   transferTaxes = Taxes(_marketing, _liquidity,  _developer);
702   require((_marketing + _liquidity + _developer) <= 99, "Must keep fees at 99% or
less");
703   }
704
705   function updateRouterAndPair(address newRouter, address newPair) external onlyOwner
{
706
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 729

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
728   function bulkExemptFee(address[] memory accounts, bool state) external onlyOwner {
729   for (uint256 i = 0; i < accounts.length; i++) {
730   exemptFee[accounts[i]] = state;
731   }
732   }
733
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 742

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
741   require(maxWallet >= 10_000, "Cannot set max wallet amount lower than 1%");
742   maxBuyLimit = maxBuy * 10**decimals();
743   maxSellLimit = maxSell * 10**decimals();
744   maxWalletLimit = maxWallet * 10**decimals();
745   }
746
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 742

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
741   require(maxWallet >= 10_000, "Cannot set max wallet amount lower than 1%");
742   maxBuyLimit = maxBuy * 10**decimals();
743   maxSellLimit = maxSell * 10**decimals();
744   maxWalletLimit = maxWallet * 10**decimals();
745   }
746
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 743

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
742   maxBuyLimit = maxBuy * 10**decimals();
743   maxSellLimit = maxSell * 10**decimals();
744   maxWalletLimit = maxWallet * 10**decimals();
745   }
746
747
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 743

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
742   maxBuyLimit = maxBuy * 10**decimals();
743   maxSellLimit = maxSell * 10**decimals();
744   maxWalletLimit = maxWallet * 10**decimals();
745   }
746
747
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 744

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
743   maxSellLimit = maxSell * 10**decimals();
744   maxWalletLimit = maxWallet * 10**decimals();
745   }
746
747   function rescueBNB(uint256 weiAmount) external onlyOwner {
748
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 744

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
743   maxSellLimit = maxSell * 10**decimals();
744   maxWalletLimit = maxWallet * 10**decimals();
745   }
746
747   function rescueBNB(uint256 weiAmount) external onlyOwner {
748
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 17

low SEVERITY

The current pragma Solidity directive is ""^0.8.17"". 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

- CustodiyV3.sol

Locations

```
16
17   pragma solidity ^0.8.17;
18
19   abstract contract Context {
20   function _msgSender() internal view virtual returns (address) {
21
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 637

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
636   address[] memory path = new address[](2);
637   path[0] = address(this);
638   path[1] = router.WETH();
639
640   _approve(address(this), address(router), tokenAmount);
641
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 638

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
637   path[0] = address(this);
638   path[1] = router.WETH();
639
640   _approve(address(this), address(router), tokenAmount);
641
642
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 730

low SEVERITY

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

Source File

- CustodiyV3.sol

Locations

```
729   for (uint256 i = 0; i < accounts.length; i++) {
730   exemptFee[accounts[i]] = state;
731   }
732   }
733
734
```

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

LINE 512

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

- CustodiyV3.sol

Locations

```
511
512   if(block.number < genesis_block + 50 && sender == pair) {
513   nonCustodial[recipient] = true;
514   }
515
516
```

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

LINE 714

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

- CustodiyV3.sol

Locations

```
713   providingLiquidity = true;
714   genesis_block = block.number;
715   }
716
717   function updateWallets(address _marketingWallet, address _devWallet) external
onlyOwner {
718
```

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.