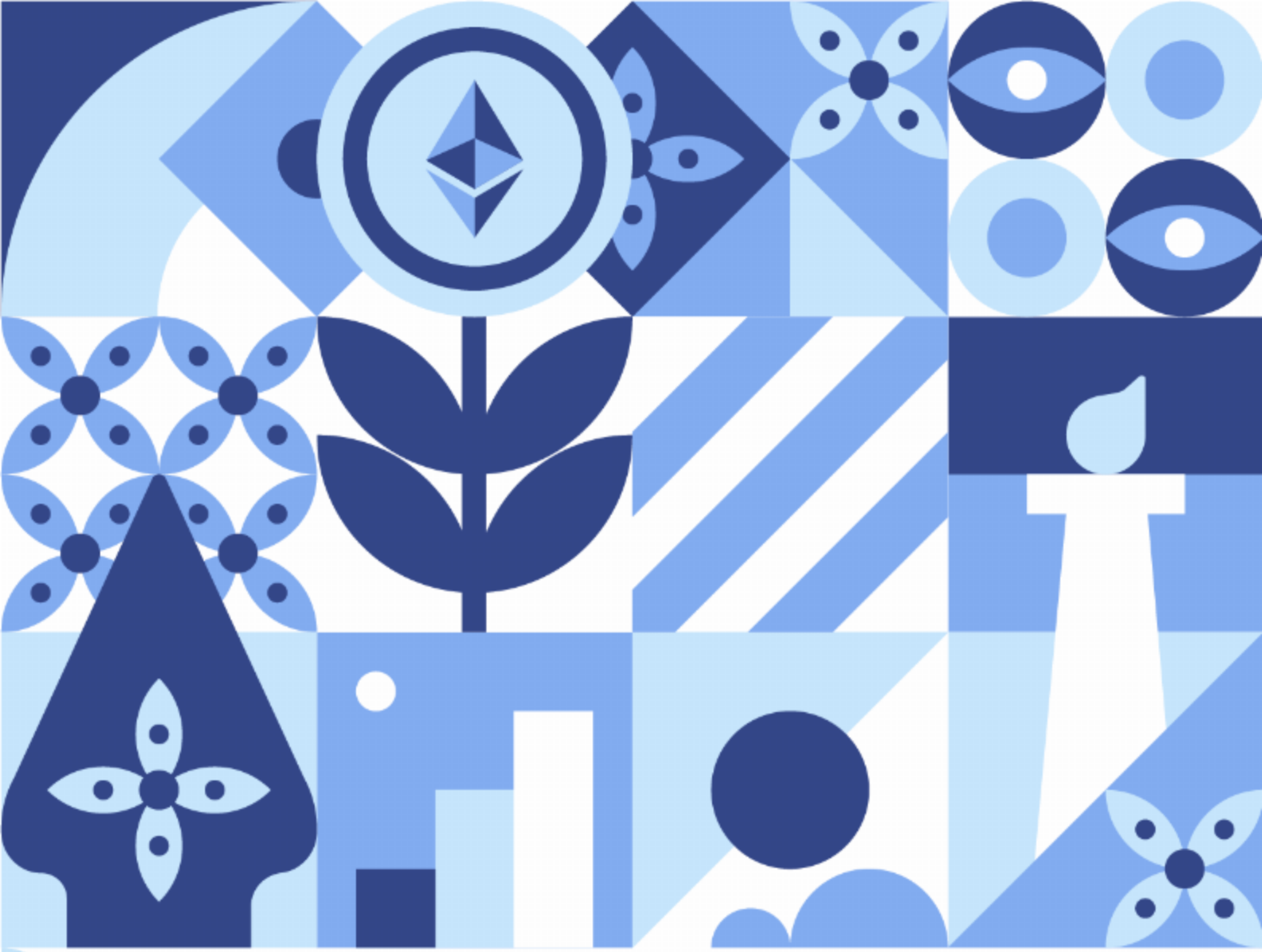

Ryukyu

Smart Contract Audit Report

SYSFIXED

09 Dec 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Ryukyu	RYU	Ethereum

Addresses

Contract address	0xce81cf156dbd2d8f4e63edc6065740affdde66e9
Contract deployer address	0xb68e5D12da294EC4Ef9fdBaDAB4B1B015F946714

Project Website

https://ryukyuerc.com/

Codebase

https://etherscan.io/address/0xce81cf156dbd2d8f4e63edc6065740affdde66e9#code

SUMMARY

$RYU is a tough dog for a tough community, if you are willing to hold through the tough times, you will reap the real rewards for it. $RYU is also loyal, that is the way we aim to be as a team towards our community. We are constantly building and developing $RYU, with plans of making it the next blue chip dog coin. With open arms, we invite anybody and everybody to join the Ryukyu army. To those listening, if you have a creative mind and need a medium to express yourself, if you have connections in the space, if you are looking for something to build, if you are looking for people you can talk to and trust, join Ryukyu.

Contract Summary

Documentation Quality

Ryukyu 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 Ryukyu 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 395, 414, 436, 469, 471, 492, 493, 518, 520, 616, 630, 645, 646, 659, 671, 686, 700, 714, 728, 744, 767, 790, 816, 1150, 1152, 1153, 1154, 1154, 1159, 1159, 1164, 1164, 1213, 1213, 1217, 1217, 1226, 1226, 1226, 1229, 1229, 1234, 1234, 1234, 1237, 1237, 1260, 1260, 1272, 1272, 1372, 1387, 1417, 1436, 1436, 1436, 1437, 1437, 1437, 1438, 1438, 1438, 1443, 1443, 1443, 1444, 1444, 1444, 1445, 1445, 1445, 1452, 1493, 1493, 1502, 1503, 1507, 1507, 1507, 1523, 1523 and 1592.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 19.
- 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 1461 and 1462.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 1354 and 1358.

- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 1355 and 1358.

CONCLUSION

We have audited the Ryukyu project released on December 2022 to discover issues and identify potential security vulnerabilities in Ryukyu 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 Ryukyu 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, 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
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.	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
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	Thursday Dec 08 2022 11:58:25 GMT+0000 (Coordinated Universal Time)
Finished	Friday Dec 09 2022 15:22:51 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	Ryukyu.sol

Detected Issues

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

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

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

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

SWC-101	ARITHMETIC OPERATION "-=" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "+" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "/" DISCOVERED	low	acknowledged
SWC-101	ARITHMETIC OPERATION "*" DISCOVERED	low	acknowledged
SWC-101	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-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-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 395

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
394   unchecked {
395   _approve(sender, _msgSender(), currentAllowance - amount);
396   }
397
398   return true;
399
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 414

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
413   function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
414   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
415   return true;
416   }
417
418
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 436

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
435   unchecked {
436   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
437   }
438
439   return true;
440
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 469

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
468   unchecked {
469   _balances[sender] = senderBalance - amount;
470   }
471   _balances[recipient] += amount;
472
473
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 471

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
470   }
471   _balances[recipient] += amount;
472
473   emit Transfer(sender, recipient, amount);
474
475
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 492

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
491
492   _totalSupply += amount;
493   _balances[account] += amount;
494   emit Transfer(address(0), account, amount);
495
496
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 493

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
492   _totalSupply += amount;
493   _balances[account] += amount;
494   emit Transfer(address(0), account, amount);
495
496   _afterTokenTransfer(address(0), account, amount);
497
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 518

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
517   unchecked {
518   _balances[account] = accountBalance - amount;
519   }
520   _totalSupply -= amount;
521
522
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 520

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
519   }
520   _totalSupply -= amount;
521
522   emit Transfer(account, address(0), amount);
523
524
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 616

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
615   unchecked {
616   uint256 c = a + b;
617   if (c < a) return (false, 0);
618   return (true, c);
619   }
620
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 630

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
629   if (b > a) return (false, 0);
630   return (true, a - b);
631   }
632   }
633
634
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 645

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
644   if (a == 0) return (true, 0);
645   uint256 c = a * b;
646   if (c / a != b) return (false, 0);
647   return (true, c);
648   }
649
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 646

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
645   uint256 c = a * b;
646   if (c / a != b) return (false, 0);
647   return (true, c);
648   }
649   }
650
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 659

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
658   if (b == 0) return (false, 0);
659   return (true, a / b);
660   }
661   }
662
663
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 671

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
670   if (b == 0) return (false, 0);
671   return (true, a % b);
672   }
673   }
674
675
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 686

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
685   function add(uint256 a, uint256 b) internal pure returns (uint256) {
686   return a + b;
687   }
688
689   /**
690
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 700

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
699   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
700   return a - b;
701   }
702
703   /**
704
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 714

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
713   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
714   return a * b;
715   }
716
717   /**
718
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 728

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
727   function div(uint256 a, uint256 b) internal pure returns (uint256) {
728   return a / b;
729   }
730
731   /**
732
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 744

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
743   function mod(uint256 a, uint256 b) internal pure returns (uint256) {
744   return a % b;
745   }
746
747   /**
748
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 767

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
766   require(b <= a, errorMessage);
767   return a - b;
768   }
769   }
770
771
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 790

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
789   require(b > 0, errorMessage);
790   return a / b;
791   }
792   }
793
794
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 816

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
815   require(b > 0, errorMessage);
816   return a % b;
817   }
818   }
819   }
820
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1150

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1149
1150   uint256 totalSupply = 1_000_000_000 * 1e18;
1151
1152   maxTransactionAmount = 20_000_000 * 1e18; // 2% from total supply
maxTransactionAmountTxn
1153   maxWallet = 20_000_000 * 1e18; // 3% from total supply maxWallet
1154
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1152

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1151
1152   maxTransactionAmount = 20_000_000 * 1e18; // 2% from total supply
maxTransactionAmountTxn
1153   maxWallet = 20_000_000 * 1e18; // 3% from total supply maxWallet
1154   swapTokensAtAmount = (totalSupply * 10) / 10000; // 0.1% swap wallet
1155
1156
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1153

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1152   maxTransactionAmount = 20_000_000 * 1e18; // 2% from total supply
maxTransactionAmountTxn
1153   maxWallet = 20_000_000 * 1e18; // 3% from total supply maxWallet
1154   swapTokensAtAmount = (totalSupply * 10) / 10000; // 0.1% swap wallet
1155
1156   buyMarketingFee = _buyMarketingFee;
1157
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1154

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1153   maxWallet = 20_000_000 * 1e18; // 3% from total supply maxWallet
1154   swapTokensAtAmount = (totalSupply * 10) / 10000; // 0.1% swap wallet
1155
1156   buyMarketingFee = _buyMarketingFee;
1157   buyLiquidityFee = _buyLiquidityFee;
1158
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1154

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1153   maxWallet = 20_000_000 * 1e18; // 3% from total supply maxWallet
1154   swapTokensAtAmount = (totalSupply * 10) / 10000; // 0.1% swap wallet
1155
1156   buyMarketingFee = _buyMarketingFee;
1157   buyLiquidityFee = _buyLiquidityFee;
1158
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1159

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1158   buyDevFee = _buyDevFee;
1159   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1160
1161   sellMarketingFee = _sellMarketingFee;
1162   sellLiquidityFee = _sellLiquidityFee;
1163
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1159

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1158   buyDevFee = _buyDevFee;
1159   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1160
1161   sellMarketingFee = _sellMarketingFee;
1162   sellLiquidityFee = _sellLiquidityFee;
1163
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1164

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1163   sellDevFee = _sellDevFee;
1164   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1165
1166   marketingWallet = address(0xb68e5D12da294EC4Ef9fdBaDAB4B1B015F946714); // set as
marketing wallet
1167   devWallet = address(0xb68e5D12da294EC4Ef9fdBaDAB4B1B015F946714); // set as dev
wallet
1168
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1164

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1163   sellDevFee = _sellDevFee;
1164   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1165
1166   marketingWallet = address(0xb68e5D12da294EC4Ef9fdBaDAB4B1B015F946714); // set as
marketing wallet
1167   devWallet = address(0xb68e5D12da294EC4Ef9fdBaDAB4B1B015F946714); // set as dev
wallet
1168
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1213

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1212   require(
1213   newAmount >= (totalSupply() * 1) / 100000,
1214   "Swap amount cannot be lower than 0.001% total supply."
1215   );
1216   require(
1217
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1213

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1212   require(
1213   newAmount >= (totalSupply() * 1) / 100000,
1214   "Swap amount cannot be lower than 0.001% total supply."
1215   );
1216   require(
1217
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1217

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1216   require(
1217   newAmount <= (totalSupply() * 5) / 1000,
1218   "Swap amount cannot be higher than 0.5% total supply."
1219   );
1220   swapTokensAtAmount = newAmount;
1221
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1217

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1216   require(
1217   newAmount <= (totalSupply() * 5) / 1000,
1218   "Swap amount cannot be higher than 0.5% total supply."
1219   );
1220   swapTokensAtAmount = newAmount;
1221
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1226

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1225   require(
1226   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1227   "Cannot set maxTransactionAmount lower than 0.1%"
1228   );
1229   maxTransactionAmount = newNum * (10**18);
1230
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1226

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1225   require(
1226   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1227   "Cannot set maxTransactionAmount lower than 0.1%"
1228   );
1229   maxTransactionAmount = newNum * (10**18);
1230
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1226

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1225   require(
1226   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1227   "Cannot set maxTransactionAmount lower than 0.1%"
1228   );
1229   maxTransactionAmount = newNum * (10**18);
1230
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1228   );
1229   maxTransactionAmount = newNum * (10**18);
1230   }
1231
1232   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1233
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1228   );
1229   maxTransactionAmount = newNum * (10**18);
1230   }
1231
1232   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1233
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1234

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1233   require(
1234   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1235   "Cannot set maxWallet lower than 0.1%"
1236   );
1237   maxWallet = newNum * (10**18);
1238
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1234

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1233   require(
1234   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1235   "Cannot set maxWallet lower than 0.1%"
1236   );
1237   maxWallet = newNum * (10**18);
1238
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1234

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1233   require(
1234   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1235   "Cannot set maxWallet lower than 0.1%"
1236   );
1237   maxWallet = newNum * (10**18);
1238
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1237

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1236   );
1237   maxWallet = newNum * (10**18);
1238   }
1239
1240   function excludeFromMaxTransaction(address updAds, bool isEx)
1241
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1237

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1236   );
1237   maxWallet = newNum * (10**18);
1238   }
1239
1240   function excludeFromMaxTransaction(address updAds, bool isEx)
1241
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1260

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1259   buyDevFee = _devFee;
1260   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1261   require(buyTotalFees <= 40, "Must keep fees at 40% or less");
1262   }
1263
1264
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1260

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1259   buyDevFee = _devFee;
1260   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1261   require(buyTotalFees <= 40, "Must keep fees at 40% or less");
1262   }
1263
1264
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1272

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1271   sellDevFee = _devFee;
1272   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1273   require(sellTotalFees <= 40, "Must keep fees at 40% or less");
1274   }
1275
1276
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1272

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1271   sellDevFee = _devFee;
1272   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1273   require(sellTotalFees <= 40, "Must keep fees at 40% or less");
1274   }
1275
1276
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1372

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1371   require(
1372   amount + balanceOf(to) <= maxWallet,
1373   "Max wallet exceeded"
1374   );
1375   }
1376
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1387

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1386   require(
1387   amount + balanceOf(to) <= maxWallet,
1388   "Max wallet exceeded"
1389   );
1390   }
1391
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1417

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1416   lpBurnEnabled &&
1417   block.timestamp >= lastLpBurnTime + lpBurnFrequency &&
1418   !_isExcludedFromFees[from]
1419   ) {
1420   autoBurnLiquidityPairTokens();
1421
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1436

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1435   fees = amount.mul(sellTotalFees).div(100);
1436   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1436

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1435   fees = amount.mul(sellTotalFees).div(100);
1436   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1436

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1435   fees = amount.mul(sellTotalFees).div(100);
1436   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1437

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1436   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440   // on buy
1441
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1437

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1436   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440   // on buy
1441
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1437

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1436   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440   // on buy
1441
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440   // on buy
1441   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1442
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440   // on buy
1441   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1442
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1437   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1438   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1439   }
1440   // on buy
1441   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1442
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1443

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1442   fees = amount.mul(buyTotalFees).div(100);
1443   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1443

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1442   fees = amount.mul(buyTotalFees).div(100);
1443   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1443

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1442   fees = amount.mul(buyTotalFees).div(100);
1443   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1444

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1443   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
1448
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1444

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1443   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
1448
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1444

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1443   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
1448
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1445

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
1448   if (fees > 0) {
1449
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1445

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
1448   if (fees > 0) {
1449
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1445

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1444   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1445   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1446   }
1447
1448   if (fees > 0) {
1449
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1452

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1451
1452   amount -= fees;
1453   }
1454
1455   super._transfer(from, to, amount);
1456
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1493

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1492   uint256 contractBalance = balanceOf(address(this));
1493   uint256 totalTokensToSwap = tokensForLiquidity +
1494   tokensForMarketing +
1495   tokensForDev;
1496   bool success;
1497
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1493

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1492   uint256 contractBalance = balanceOf(address(this));
1493   uint256 totalTokensToSwap = tokensForLiquidity +
1494   tokensForMarketing +
1495   tokensForDev;
1496   bool success;
1497
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1502

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1501
1502   if (contractBalance > swapTokensAtAmount * 20) {
1503   contractBalance = swapTokensAtAmount * 20;
1504   }
1505
1506
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1503

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1502   if (contractBalance > swapTokensAtAmount * 20) {
1503   contractBalance = swapTokensAtAmount * 20;
1504   }
1505
1506   // Halve the amount of liquidity tokens
1507
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1507

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1506   // Halve the amount of liquidity tokens
1507   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1508   totalTokensToSwap /
1509   2;
1510   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1511
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1507

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1506   // Halve the amount of liquidity tokens
1507   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1508   totalTokensToSwap /
1509   2;
1510   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1511
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1507

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1506   // Halve the amount of liquidity tokens
1507   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1508   totalTokensToSwap /
1509   2;
1510   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1511
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1523

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1522
1523   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1524
1525   tokensForLiquidity = 0;
1526   tokensForMarketing = 0;
1527
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1523

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1522
1523   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1524
1525   tokensForLiquidity = 0;
1526   tokensForMarketing = 0;
1527
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1592

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1591   require(
1592   block.timestamp > lastManualLpBurnTime + manualBurnFrequency,
1593   "Must wait for cooldown to finish"
1594   );
1595   require(percent <= 1000, "May not nuke more than 10% of tokens in LP");
1596
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 19

low SEVERITY

The current pragma Solidity directive is ""=0.8.10>=0.8.10>=0.8.0<0.9.0"". 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

- Ryukyu.sol

Locations

```
18   // SPDX-License-Identifier: MIT
19   pragma solidity =0.8.10 >=0.8.10 >=0.8.0 <0.9.0;
20   pragma experimental ABIEncoderV2;
21
22   ////// lib/openzeppelin-contracts/contracts/utils/Context.sol
23
```

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

LINE 1354

low SEVERITY

The tx.origin environment variable has been found to influence a control flow decision. Note that using "tx.origin" as a security control might cause a situation where a user inadvertently authorizes a smart contract to perform an action on their behalf. It is recommended to use "msg.sender" instead.

Source File

- Ryukyu.sol

Locations

```
1353   require(
1354   _holderLastTransferTimestamp[tx.origin] <
1355   block.number,
1356   "_transfer:: Transfer Delay enabled.  Only one purchase per block allowed."
1357   );
1358
```

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

LINE 1358

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

- Ryukyu.sol

Locations

```
1357   );
1358   _holderLastTransferTimestamp[tx.origin] = block.number;
1359   }
1360   }
1361
1362
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1461

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1460   address[] memory path = new address[](2);
1461   path[0] = address(this);
1462   path[1] = uniswapV2Router.WETH();
1463
1464   _approve(address(this), address(uniswapV2Router), tokenAmount);
1465
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1462

low SEVERITY

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

Source File

- Ryukyu.sol

Locations

```
1461   path[0] = address(this);
1462   path[1] = uniswapV2Router.WETH();
1463
1464   _approve(address(this), address(uniswapV2Router), tokenAmount);
1465
1466
```

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

LINE 1355

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

- Ryukyu.sol

Locations

```
1354   _holderLastTransferTimestamp[tx.origin] <
1355   block.number,
1356   "_transfer:: Transfer Delay enabled.  Only one purchase per block allowed."
1357   );
1358   _holderLastTransferTimestamp[tx.origin] = block.number;
1359
```

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

LINE 1358

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

- Ryukyu.sol

Locations

```
1357   );
1358   _holderLastTransferTimestamp[tx.origin] = block.number;
1359   }
1360   }
1361
1362
```

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.