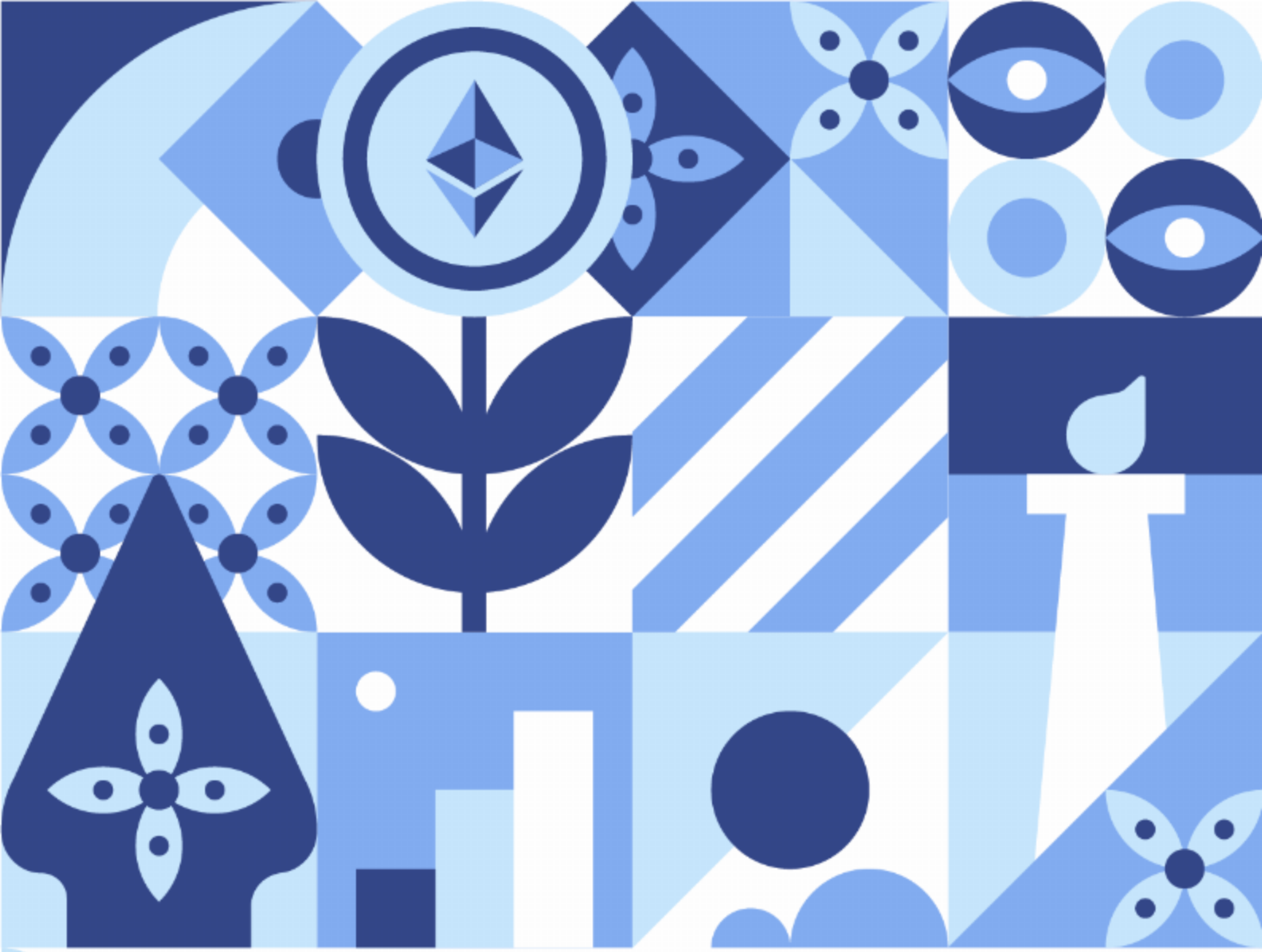

Game Of Dragons

Smart Contract Audit Report

SYSFIXED

25 Aug 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Game Of Dragons	$GOD	Ethereum

Addresses

Contract address	0x2F60EbD82577e95B8f792988D414032b46271c1c
Contract deployer address	0x50b43abe17C5466659e78Ce3902e24c4ddFA8316

Project Website

https://www.game-of-dragons.com/

Codebase

https://etherscan.io/address/0x2F60EbD82577e95B8f792988D414032b46271c1c#code

SUMMARY

Game of Dragons is a third person dragon fighting MMO where both investors and gamers come together! Fight EPIC battle against other players. Race with your dragons against other players. Breed and collect rare NFTs, Stake tokens for massive rewards, Play2Earn while you breath fire on your enemy's. Collect abilities and items. Trade on the NFT Marketplace. Play together with friends and collect achievements!

Contract Summary

Documentation Quality

Game Of Dragons 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 Game Of Dragons with the discovery of several low issues.

Test Coverage

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

Audit Findings Summary

- SWC-100 SWC-108 | Explicitly define visibility for all state variables on lines 470.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 137, 147, 155, 174, 176, 188, 189, 203, 205, 477, 477, 478, 478, 541, 541, 542, 656, 656, 699, 723, 762, 762, 762, 763, 763, 763, 765, 765, 766, 766, 768, 768, 783, 787, 787, 801, 801, 807, 807, 832, 832, 833, 845, 845, 846, 861, 885, 885, 912, 912, 912, 913, 913, 913, 916, 916, 917 and 917.
- 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 771 and 772.

CONCLUSION

We have audited the Game Of Dragons project released on August 2022 to discover issues and identify potential security vulnerabilities in Game Of Dragons 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 Game Of Dragons 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 state variable visibility is not set 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.	ISSUE FOUND
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	PASS
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.	PASS
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 Aug 24 2022 14:58:33 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Aug 25 2022 17:11:38 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	GameOfDragons.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-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 137

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
136   unchecked {
137   _approve(sender, _msgSender(), currentAllowance - amount);
138   }
139   }
140
141
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 147

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
146   function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
147   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
148   return true;
149   }
150
151
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 155

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
154   unchecked {
155   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
156   }
157
158   return true;
159
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 174

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
173   unchecked {
174   _balances[sender] = senderBalance - amount;
175   }
176   _balances[recipient] += amount;
177
178
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 176

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
175   }
176   _balances[recipient] += amount;
177
178   emit Transfer(sender, recipient, amount);
179
180
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 188

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
187
188   _totalSupply += amount;
189   _balances[account] += amount;
190   emit Transfer(address(0), account, amount);
191
192
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 189

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
188   _totalSupply += amount;
189   _balances[account] += amount;
190   emit Transfer(address(0), account, amount);
191
192   _afterTokenTransfer(address(0), account, amount);
193
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 203

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
202   unchecked {
203   _balances[account] = accountBalance - amount;
204   }
205   _totalSupply -= amount;
206
207
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 205

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
204   }
205   _totalSupply -= amount;
206
207   emit Transfer(account, address(0), amount);
208
209
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 477

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
476   bool    public  maxTransactionLimitEnabled = true;
477   uint256 public  maxTransactionAmountBuy  = 5 * (10**23); //0.5% of total supply
478   uint256 public  maxTransactionAmountSell = 5 * (10**23); //0.5% of total supply
479
480   event ExcludeFromFees(address indexed account, bool isExcluded);
481
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 477

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
476   bool    public  maxTransactionLimitEnabled = true;
477   uint256 public  maxTransactionAmountBuy  = 5 * (10**23); //0.5% of total supply
478   uint256 public  maxTransactionAmountSell = 5 * (10**23); //0.5% of total supply
479
480   event ExcludeFromFees(address indexed account, bool isExcluded);
481
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 478

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
477   uint256 public  maxTransactionAmountBuy  = 5 * (10**23); //0.5% of total supply
478   uint256 public  maxTransactionAmountSell = 5 * (10**23); //0.5% of total supply
479
480   event ExcludeFromFees(address indexed account, bool isExcluded);
481   event FeesUpdated(uint256 buyFee, uint256 sellFee);
482
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 478

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
477   uint256 public  maxTransactionAmountBuy  = 5 * (10**23); //0.5% of total supply
478   uint256 public  maxTransactionAmountSell = 5 * (10**23); //0.5% of total supply
479
480   event ExcludeFromFees(address indexed account, bool isExcluded);
481   event FeesUpdated(uint256 buyFee, uint256 sellFee);
482
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 541

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
540
541   _mint(newOwner, 100000000 * (10**18));
542   swapTokensAtAmount = totalSupply() / 2000;
543
544   operator = _msgSender();
545
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 541

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
540
541   _mint(newOwner, 100000000 * (10**18));
542   swapTokensAtAmount = totalSupply() / 2000;
543
544   operator = _msgSender();
545
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 542

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
541   _mint(newOwner, 100000000 * (10**18));
542   swapTokensAtAmount = totalSupply() / 2000;
543
544   operator = _msgSender();
545   }
546
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 656

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
655   function updateFeeShares(uint256 _fee1FeeShare, uint256 _liquidityFeeShare, uint256
_fee2Share) external onlyOwner {
656   require(_fee1FeeShare + _liquidityFeeShare + _fee2Share == 100, "Fee shares must
add up to 100");
657   fee1Share = _fee1FeeShare;
658   liquidityShare = _liquidityFeeShare;
659   fee2Share = _fee2Share;
660
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 656

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
655   function updateFeeShares(uint256 _fee1FeeShare, uint256 _liquidityFeeShare, uint256
_fee2Share) external onlyOwner {
656   require(_fee1FeeShare + _liquidityFeeShare + _fee2Share == 100, "Fee shares must
add up to 100");
657   fee1Share = _fee1FeeShare;
658   liquidityShare = _liquidityFeeShare;
659   fee2Share = _fee2Share;
660
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 699

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
698   if (launchTime > 0) {
699   if(block.timestamp - launchTime <= timeAntiBot && from == uniswapV2Pair &&
antibotSystemEnable) {
700
701   _isBot[to] = true;
702   }
703
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
722   uint balance  = balanceOf(to);
723   require(balance + amount <= maxWalletAmount(), "MaxWallet: Transfer amount exceeds
the maxWalletAmount");
724   }
725   }
726
727
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 762

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
761
762   uint256 liquidityTokens       = contractTokenBalance * liquidityShare / 100 / 2;
763   uint256 liquidityTokensForETH = contractTokenBalance * liquidityShare / 100 / 2;
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 762

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
761
762   uint256 liquidityTokens       = contractTokenBalance * liquidityShare / 100 / 2;
763   uint256 liquidityTokensForETH = contractTokenBalance * liquidityShare / 100 / 2;
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 762

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
761
762   uint256 liquidityTokens       = contractTokenBalance * liquidityShare / 100 / 2;
763   uint256 liquidityTokensForETH = contractTokenBalance * liquidityShare / 100 / 2;
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 763

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
762   uint256 liquidityTokens       = contractTokenBalance * liquidityShare / 100 / 2;
763   uint256 liquidityTokensForETH = contractTokenBalance * liquidityShare / 100 / 2;
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766   uint256 fee2Tokens = contractTokenBalance * fee2Share / 100;
767
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 763

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
762   uint256 liquidityTokens       = contractTokenBalance * liquidityShare / 100 / 2;
763   uint256 liquidityTokensForETH = contractTokenBalance * liquidityShare / 100 / 2;
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766   uint256 fee2Tokens = contractTokenBalance * fee2Share / 100;
767
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 763

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
762   uint256 liquidityTokens       = contractTokenBalance * liquidityShare / 100 / 2;
763   uint256 liquidityTokensForETH = contractTokenBalance * liquidityShare / 100 / 2;
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766   uint256 fee2Tokens = contractTokenBalance * fee2Share / 100;
767
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 765

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766   uint256 fee2Tokens = contractTokenBalance * fee2Share / 100;
767
768   uint256 tokensToSwap  = liquidityTokensForETH + fee1Tokens + fee2Tokens;
769
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 765

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
764
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766   uint256 fee2Tokens = contractTokenBalance * fee2Share / 100;
767
768   uint256 tokensToSwap  = liquidityTokensForETH + fee1Tokens + fee2Tokens;
769
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 766

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766   uint256 fee2Tokens = contractTokenBalance * fee2Share / 100;
767
768   uint256 tokensToSwap  = liquidityTokensForETH + fee1Tokens + fee2Tokens;
769
770
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 766

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
765   uint256 fee1Tokens = contractTokenBalance * fee1Share / 100;
766   uint256 fee2Tokens = contractTokenBalance * fee2Share / 100;
767
768   uint256 tokensToSwap  = liquidityTokensForETH + fee1Tokens + fee2Tokens;
769
770
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 768

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
767
768   uint256 tokensToSwap  = liquidityTokensForETH + fee1Tokens + fee2Tokens;
769
770   address[] memory path = new address[](2);
771   path[0] = address(this);
772
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 768

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
767
768   uint256 tokensToSwap  = liquidityTokensForETH + fee1Tokens + fee2Tokens;
769
770   address[] memory path = new address[](2);
771   path[0] = address(this);
772
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 783

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
782
783   uint256 newBalance = address(this).balance - initialBalance;
784
785   if (liquidityShare > 0)
786   {
787
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 787

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
786   {
787   uint256 liquidityETH = newBalance * liquidityTokensForETH / tokensToSwap;
788
789   uniswapV2Router.addLiquidityETH{value: liquidityETH}(
790   address(this),
791
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 787

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
786   {
787   uint256 liquidityETH = newBalance * liquidityTokensForETH / tokensToSwap;
788
789   uniswapV2Router.addLiquidityETH{value: liquidityETH}(
790   address(this),
791
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 801

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
800   if(fee1Share > 0) {
801   uint256 fee1ETH = newBalance * fee1Tokens / tokensToSwap;
802   sendETH(payable(fee1Wallet), fee1ETH);
803
804   }
805
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 801

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
800   if(fee1Share > 0) {
801   uint256 fee1ETH = newBalance * fee1Tokens / tokensToSwap;
802   sendETH(payable(fee1Wallet), fee1ETH);
803
804   }
805
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 807

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
806   if(fee2Share > 0) {
807   uint256 fee2ETH = newBalance * fee2Tokens / tokensToSwap;
808   sendETH(payable(fee2Wallet), fee2ETH);
809
810   }
811
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 807

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
806   if(fee2Share > 0) {
807   uint256 fee2ETH = newBalance * fee2Tokens / tokensToSwap;
808   sendETH(payable(fee2Wallet), fee2ETH);
809
810   }
811
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 832

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
831   _totalFees = 900;
832   uint256 fees = amount * _totalFees / 1000;
833   amount = amount - fees;
834   super._transfer(from, botFeeWallet, fees);
835   }
836
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 832

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
831   _totalFees = 900;
832   uint256 fees = amount * _totalFees / 1000;
833   amount = amount - fees;
834   super._transfer(from, botFeeWallet, fees);
835   }
836
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 833

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
832   uint256 fees = amount * _totalFees / 1000;
833   amount = amount - fees;
834   super._transfer(from, botFeeWallet, fees);
835   }
836
837
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 845

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
844   }
845   uint256 fees = amount * _totalFees / 1000;
846   amount = amount - fees;
847   super._transfer(from, address(this), fees);
848   }
849
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 845

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
844   }
845   uint256 fees = amount * _totalFees / 1000;
846   amount = amount - fees;
847   super._transfer(from, address(this), fees);
848   }
849
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 846

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
845   uint256 fees = amount * _totalFees / 1000;
846   amount = amount - fees;
847   super._transfer(from, address(this), fees);
848   }
849
850
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 861

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
860   function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
861   require(newAmount > totalSupply() / 100000, "SwapTokensAtAmount must be greater
than 0.001% of total supply");
862   swapTokensAtAmount = newAmount;
863   }
864
865
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 885

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
884   function maxWalletAmount() public view returns (uint256) {
885   return totalSupply() * maxWalletLimitRate / 1000;
886   }
887
888   function setMaxWalletRate_Denominator1000(uint256 _val) external onlyOwner {
889
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 885

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
884   function maxWalletAmount() public view returns (uint256) {
885   return totalSupply() * maxWalletLimitRate / 1000;
886   }
887
888   function setMaxWalletRate_Denominator1000(uint256 _val) external onlyOwner {
889
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 912

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
911   require(
912   _maxTransactionAmountBuy  >= totalSupply() / (10 ** decimals()) / 1000 &&
913   _maxTransactionAmountSell >= totalSupply() / (10 ** decimals()) / 1000,
914   "Max Transaction limis cannot be lower than 0.1% of total supply"
915   );
916
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 912

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
911   require(
912   _maxTransactionAmountBuy  >= totalSupply() / (10 ** decimals()) / 1000 &&
913   _maxTransactionAmountSell >= totalSupply() / (10 ** decimals()) / 1000,
914   "Max Transaction limis cannot be lower than 0.1% of total supply"
915   );
916
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 912

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
911   require(
912   _maxTransactionAmountBuy  >= totalSupply() / (10 ** decimals()) / 1000 &&
913   _maxTransactionAmountSell >= totalSupply() / (10 ** decimals()) / 1000,
914   "Max Transaction limis cannot be lower than 0.1% of total supply"
915   );
916
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 913

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
912   _maxTransactionAmountBuy  >= totalSupply() / (10 ** decimals()) / 1000 &&
913   _maxTransactionAmountSell >= totalSupply() / (10 ** decimals()) / 1000,
914   "Max Transaction limis cannot be lower than 0.1% of total supply"
915   );
916   maxTransactionAmountBuy  = _maxTransactionAmountBuy  * (10 ** decimals());
917
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 913

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
912   _maxTransactionAmountBuy  >= totalSupply() / (10 ** decimals()) / 1000 &&
913   _maxTransactionAmountSell >= totalSupply() / (10 ** decimals()) / 1000,
914   "Max Transaction limis cannot be lower than 0.1% of total supply"
915   );
916   maxTransactionAmountBuy  = _maxTransactionAmountBuy  * (10 ** decimals());
917
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 913

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
912   _maxTransactionAmountBuy  >= totalSupply() / (10 ** decimals()) / 1000 &&
913   _maxTransactionAmountSell >= totalSupply() / (10 ** decimals()) / 1000,
914   "Max Transaction limis cannot be lower than 0.1% of total supply"
915   );
916   maxTransactionAmountBuy  = _maxTransactionAmountBuy  * (10 ** decimals());
917
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 916

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
915   );
916   maxTransactionAmountBuy  = _maxTransactionAmountBuy  * (10 ** decimals());
917   maxTransactionAmountSell = _maxTransactionAmountSell * (10 ** decimals());
918   emit MaxTransactionLimitRatesChanged(maxTransactionAmountBuy,
maxTransactionAmountSell);
919   }
920
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 916

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
915   );
916   maxTransactionAmountBuy  = _maxTransactionAmountBuy  * (10 ** decimals());
917   maxTransactionAmountSell = _maxTransactionAmountSell * (10 ** decimals());
918   emit MaxTransactionLimitRatesChanged(maxTransactionAmountBuy,
maxTransactionAmountSell);
919   }
920
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 917

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
916   maxTransactionAmountBuy  = _maxTransactionAmountBuy  * (10 ** decimals());
917   maxTransactionAmountSell = _maxTransactionAmountSell * (10 ** decimals());
918   emit MaxTransactionLimitRatesChanged(maxTransactionAmountBuy,
maxTransactionAmountSell);
919   }
920
921
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 917

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
916   maxTransactionAmountBuy  = _maxTransactionAmountBuy  * (10 ** decimals());
917   maxTransactionAmountSell = _maxTransactionAmountSell * (10 ** decimals());
918   emit MaxTransactionLimitRatesChanged(maxTransactionAmountBuy,
maxTransactionAmountSell);
919   }
920
921
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 470

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
469
470   mapping(address => bool) _isBot;
471   uint256 public launchTime = 0;
472
473   bool public antibotSystemEnable = true;
474
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 771

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
770   address[] memory path = new address[](2);
771   path[0] = address(this);
772   path[1] = uniswapV2Router.WETH();
773
774   uint256 initialBalance = address(this).balance;
775
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 772

low SEVERITY

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

Source File

- GameOfDragons.sol

Locations

```
771   path[0] = address(this);
772   path[1] = uniswapV2Router.WETH();
773
774   uint256 initialBalance = address(this).balance;
775
776
```

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.