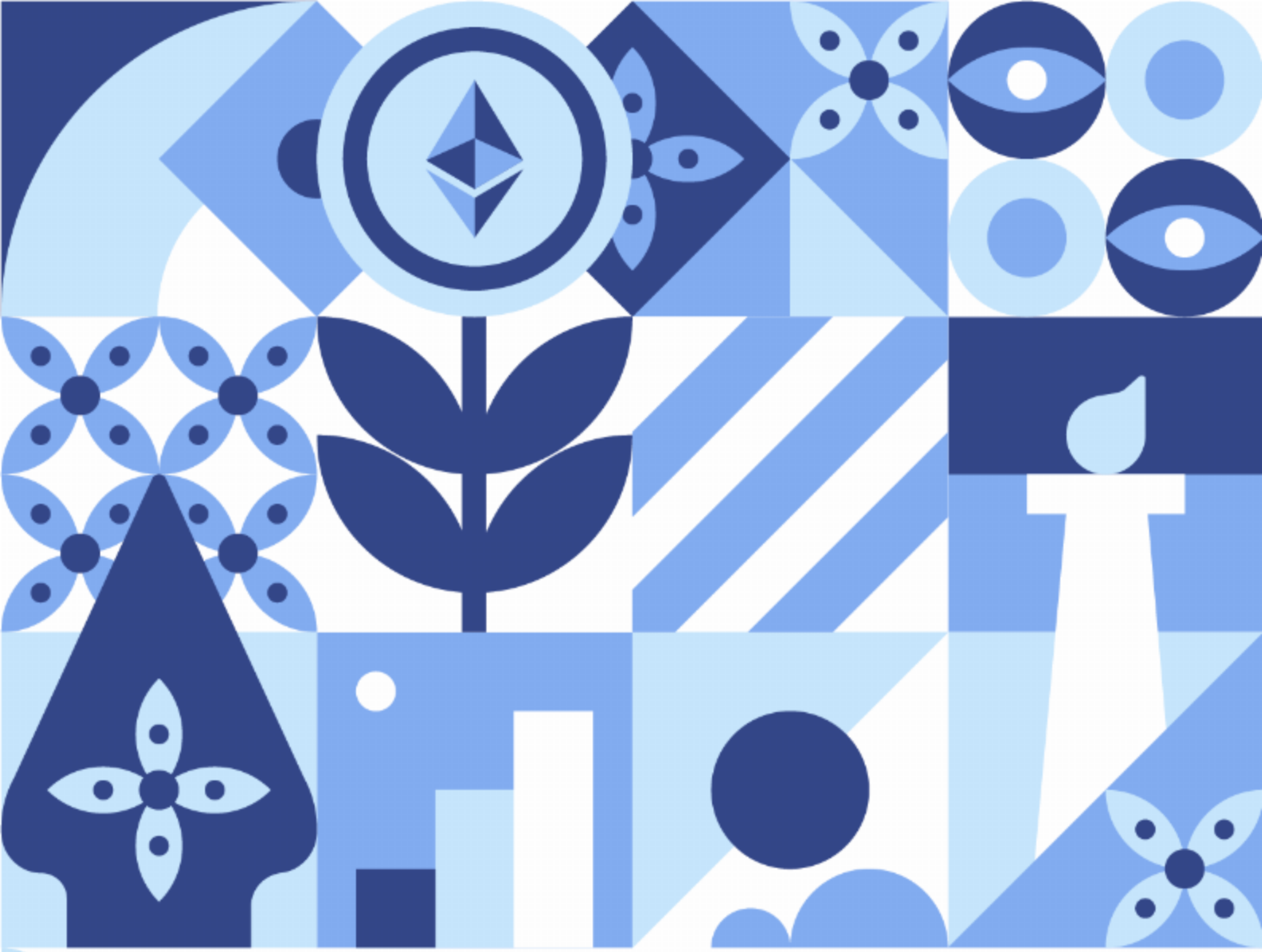

Kingdom

Smart Contract Audit Report

SYSFIXED

11 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Kingdom	KNDM	Ethereum

Addresses

Contract address	0x1ae378cc5d38350ec90ce9bcf827a544cb2bba75
Contract deployer address	0x99272926CB5995CC4eeA9Cd1Bc03BC3cD952C7De

Project Website

https://www.kingdometh.com/

Codebase

https://etherscan.io/address/0x1ae378cc5d38350ec90ce9bcf827a544cb2bba75#code

SUMMARY

The Kingdom [$KNDM] is a sustainable Medieval-based staking ERC-20 utility token that provides high APYs to its game users. The P2E/Metaverse staking ecosystem comprises of a P2E game in which users build an empire and upgrade castle elements to protect their kingdom from the opponent, collecting gold as a reward and exchanging it for Ethereum, alongside its limited NFT collection that increases user thresholds.

Contract Summary

Documentation Quality

Kingdom provides a very poor documentation with standard of solidity base code.

- The technical description is provided unclear and disorganized.

Code Quality

The Overall quality of the basecode is poor.

- Solidity basecode and rules are unclear and disorganized by Kingdom.

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 296, 297, 298 and 299.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 130, 168, 170, 189, 196, 197, 214, 338, 348, 348, 349, 349, 356, 356, 357, 357, 371, 373, 393, 393, 407, 407, 507, 522, 535, 535, 538, 540, 540, 540, 544, 562, 562, 563, 563, 564, 564, 566, 566, 568, 569, 570, 574, 574, 575, 575, 576, 576, 578, 578, 580, 581, 582, 587, 623, 623, 627, 628, 628, 628, 631, 631, 631, 637, 639, 639, 639 and 640.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 6.
- SWC-104 | It is recommended to use handle at low-level call methods on lines 643 and 647.
- 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 372, 372, 373, 373, 610 and 611.

CONCLUSION

We have audited the Kingdom project released on January 2023 to discover issues and identify potential security vulnerabilities in Kingdom 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 Kingdom smart contract code do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a floating pragma is set, a state variable visibility is not set, out of bounds array access and unchecked return value from low-level external call. We recommend If you choose to use low-level call methods, make sure to handle the possibility that the call will fail by checking the return 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.	ISSUE FOUND
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	ISSUE FOUND
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	Tuesday Jan 10 2023 07:56:20 GMT+0000 (Coordinated Universal Time)
Finished	Wednesday Jan 11 2023 18:05:40 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	KingdomToken.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-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-104	UNCHECKED RETURN VALUE FROM LOW-LEVEL EXTERNAL CALL.	medium	acknowledged
SWC-104	UNCHECKED RETURN VALUE FROM LOW-LEVEL EXTERNAL CALL.	medium	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 130

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
129   unchecked {
130   _approve(sender, msg.sender, currentAllowance - amount);
131   }
132
133   return true;
134
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 168

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
167   unchecked {
168   _balances[sender] = senderBalance - amount;
169   }
170   _balances[recipient] += amount;
171
172
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 170

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
169   }
170   _balances[recipient] += amount;
171
172   emit Transfer(sender, recipient, amount);
173
174
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 189

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
188   function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
189   _approve(msg.sender, spender, _allowances[msg.sender][spender] + addedValue);
190   return true;
191   }
192
193
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 196

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
195
196   _totalSupply += amount;
197   _balances[account] += amount;
198   emit Transfer(address(0), account, amount);
199   }
200
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 197

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
196   _totalSupply += amount;
197   _balances[account] += amount;
198   emit Transfer(address(0), account, amount);
199   }
200
201
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 214

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
213   unchecked {
214   _approve(msg.sender, spender, currentAllowance - subtractedValue);
215   }
216
217   return true;
218
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 338

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
337
338   uint256 totalSupply = 400_000_000_000 * 1e18;
339
340   buyMarketingFee = 55;
341   buyDevFee = 5;
342
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 348

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
347
348   buyTotalFees = buyMarketingFee + buyDevFee + buyLiquidityFee;
349   sellTotalFees = sellMarketingFee + sellDevFee + sellLiquidityFee;
350
351   isExcludedFromFee[address(0xdead)] = true;
352
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 348

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
347
348   buyTotalFees = buyMarketingFee + buyDevFee + buyLiquidityFee;
349   sellTotalFees = sellMarketingFee + sellDevFee + sellLiquidityFee;
350
351   isExcludedFromFee[address(0xdead)] = true;
352
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 349

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
348   buyTotalFees = buyMarketingFee + buyDevFee + buyLiquidityFee;
349   sellTotalFees = sellMarketingFee + sellDevFee + sellLiquidityFee;
350
351   isExcludedFromFee[address(0xdead)] = true;
352   isExcludedFromFee[address(this)] = true;
353
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 349

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
348   buyTotalFees = buyMarketingFee + buyDevFee + buyLiquidityFee;
349   sellTotalFees = sellMarketingFee + sellDevFee + sellLiquidityFee;
350
351   isExcludedFromFee[address(0xdead)] = true;
352   isExcludedFromFee[address(this)] = true;
353
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 356

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
355
356   maxTransactionAmount = totalSupply * 5 / 1000;
357   maxWallet = totalSupply * 10 / 1000;
358
359   /*
360
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 356

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
355
356   maxTransactionAmount = totalSupply * 5 / 1000;
357   maxWallet = totalSupply * 10 / 1000;
358
359   /*
360
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 357

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
356   maxTransactionAmount = totalSupply * 5 / 1000;
357   maxWallet = totalSupply * 10 / 1000;
358
359   /*
360   _mint is an internal function in ERC20.sol that is only called here,
361
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 357

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
356   maxTransactionAmount = totalSupply * 5 / 1000;
357   maxWallet = totalSupply * 10 / 1000;
358
359   /*
360   _mint is an internal function in ERC20.sol that is only called here,
361
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 371

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
370
371   for (uint i=0; i<holders.length; i++) {
372   super._transfer(address(this), holders[i], amounts[i]);
373   airdropAmount[holders[i]] += amounts[i];
374   }
375
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 373

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
372   super._transfer(address(this), holders[i], amounts[i]);
373   airdropAmount[holders[i]] += amounts[i];
374   }
375   }
376
377
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 393

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
392
393   buyTotalFees = buyMarketingFee + buyDevFee + buyLiquidityFee;
394
395   if (maxBuyFeeSet) {
396   require(buyTotalFees <= maxBuyFee);
397
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 393

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
392
393   buyTotalFees = buyMarketingFee + buyDevFee + buyLiquidityFee;
394
395   if (maxBuyFeeSet) {
396   require(buyTotalFees <= maxBuyFee);
397
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
406
407   sellTotalFees = sellMarketingFee + sellDevFee + sellLiquidityFee;
408
409   if (maxSellFeeSet) {
410   require(sellTotalFees <= maxSellFee);
411
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 407

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
406
407   sellTotalFees = sellMarketingFee + sellDevFee + sellLiquidityFee;
408
409   if (maxSellFeeSet) {
410   require(sellTotalFees <= maxSellFee);
411
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 507

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
506   require(
507   amount + balanceOf(to) <= maxWallet,
508   "!maxWallet"
509   );
510   }
511
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 522

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
521   require(
522   amount + balanceOf(to) <= maxWallet,
523   "!maxWallet"
524   );
525   }
526
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 535

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
534
535   uint256 elapsedPeriods = (block.timestamp - launchTime) / 86400;
536
537   if (elapsedPeriods < vestingPeriods) {
538   uint256 minimumBalance = airdroppedTokenAmount - (
539
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 535

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
534
535   uint256 elapsedPeriods = (block.timestamp - launchTime) / 86400;
536
537   if (elapsedPeriods < vestingPeriods) {
538   uint256 minimumBalance = airdroppedTokenAmount - (
539
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 538

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
537   if (elapsedPeriods < vestingPeriods) {
538   uint256 minimumBalance = airdroppedTokenAmount - (
539   // a number ranging from 0 to 100
540   elapsedPeriods * vestingPercent
541   * airdroppedTokenAmount
542
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 540

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
539   // a number ranging from 0 to 100
540   elapsedPeriods * vestingPercent
541   * airdroppedTokenAmount
542   / 100
543   );
544
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 540

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
539   // a number ranging from 0 to 100
540   elapsedPeriods * vestingPercent
541   * airdroppedTokenAmount
542   / 100
543   );
544
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 540

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
539   // a number ranging from 0 to 100
540   elapsedPeriods * vestingPercent
541   * airdroppedTokenAmount
542   / 100
543   );
544
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 544

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
543   );
544   require(balanceOf(from) - amount >= minimumBalance);
545   } else {
546   vestingFinished = true;
547   }
548
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 562

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
561   if (isAMM[to] && sellTotalFees > 0) {
562   uint256 newTokensForDev = amount * sellDevFee / feeDenominator;
563   uint256 newTokensForMarketing = amount * sellMarketingFee / feeDenominator;
564   uint256 newTokensForLiquidity = amount * sellLiquidityFee / feeDenominator;
565
566
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 562

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
561   if (isAMM[to] && sellTotalFees > 0) {
562   uint256 newTokensForDev = amount * sellDevFee / feeDenominator;
563   uint256 newTokensForMarketing = amount * sellMarketingFee / feeDenominator;
564   uint256 newTokensForLiquidity = amount * sellLiquidityFee / feeDenominator;
565
566
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 563

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
562   uint256 newTokensForDev = amount * sellDevFee / feeDenominator;
563   uint256 newTokensForMarketing = amount * sellMarketingFee / feeDenominator;
564   uint256 newTokensForLiquidity = amount * sellLiquidityFee / feeDenominator;
565
566   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
567
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 563

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
562   uint256 newTokensForDev = amount * sellDevFee / feeDenominator;
563   uint256 newTokensForMarketing = amount * sellMarketingFee / feeDenominator;
564   uint256 newTokensForLiquidity = amount * sellLiquidityFee / feeDenominator;
565
566   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
567
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 564

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
563   uint256 newTokensForMarketing = amount * sellMarketingFee / feeDenominator;
564   uint256 newTokensForLiquidity = amount * sellLiquidityFee / feeDenominator;
565
566   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
567
568
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 564

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
563   uint256 newTokensForMarketing = amount * sellMarketingFee / feeDenominator;
564   uint256 newTokensForLiquidity = amount * sellLiquidityFee / feeDenominator;
565
566   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
567
568
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 566

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
565
566   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
567
568   tokensForDev += newTokensForDev;
569   tokensForMarketing += newTokensForMarketing;
570
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 566

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
565
566   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
567
568   tokensForDev += newTokensForDev;
569   tokensForMarketing += newTokensForMarketing;
570
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 568

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
567
568   tokensForDev += newTokensForDev;
569   tokensForMarketing += newTokensForMarketing;
570   tokensForLiquidity += newTokensForLiquidity;
571   }
572
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 569

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
568   tokensForDev += newTokensForDev;
569   tokensForMarketing += newTokensForMarketing;
570   tokensForLiquidity += newTokensForLiquidity;
571   }
572
573
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 570

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
569   tokensForMarketing += newTokensForMarketing;
570   tokensForLiquidity += newTokensForLiquidity;
571   }
572
573   else if (isAMM[from] && buyTotalFees > 0) {
574
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 574

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
573   else if (isAMM[from] && buyTotalFees > 0) {
574   uint256 newTokensForDev = amount * buyDevFee / feeDenominator;
575   uint256 newTokensForMarketing = amount * buyMarketingFee / feeDenominator;
576   uint256 newTokensForLiquidity = amount * buyLiquidityFee / feeDenominator;
577
578
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 574

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
573   else if (isAMM[from] && buyTotalFees > 0) {
574   uint256 newTokensForDev = amount * buyDevFee / feeDenominator;
575   uint256 newTokensForMarketing = amount * buyMarketingFee / feeDenominator;
576   uint256 newTokensForLiquidity = amount * buyLiquidityFee / feeDenominator;
577
578
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 575

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
574   uint256 newTokensForDev = amount * buyDevFee / feeDenominator;
575   uint256 newTokensForMarketing = amount * buyMarketingFee / feeDenominator;
576   uint256 newTokensForLiquidity = amount * buyLiquidityFee / feeDenominator;
577
578   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
579
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 575

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
574   uint256 newTokensForDev = amount * buyDevFee / feeDenominator;
575   uint256 newTokensForMarketing = amount * buyMarketingFee / feeDenominator;
576   uint256 newTokensForLiquidity = amount * buyLiquidityFee / feeDenominator;
577
578   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
579
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 576

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
575   uint256 newTokensForMarketing = amount * buyMarketingFee / feeDenominator;
576   uint256 newTokensForLiquidity = amount * buyLiquidityFee / feeDenominator;
577
578   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
579
580
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 576

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
575   uint256 newTokensForMarketing = amount * buyMarketingFee / feeDenominator;
576   uint256 newTokensForLiquidity = amount * buyLiquidityFee / feeDenominator;
577
578   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
579
580
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 578

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
577
578   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
579
580   tokensForDev += newTokensForDev;
581   tokensForMarketing += newTokensForMarketing;
582
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 578

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
577
578   fees = newTokensForDev + newTokensForMarketing + newTokensForLiquidity;
579
580   tokensForDev += newTokensForDev;
581   tokensForMarketing += newTokensForMarketing;
582
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 580

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
579
580   tokensForDev += newTokensForDev;
581   tokensForMarketing += newTokensForMarketing;
582   tokensForLiquidity += newTokensForLiquidity;
583   }
584
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 581

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
580   tokensForDev += newTokensForDev;
581   tokensForMarketing += newTokensForMarketing;
582   tokensForLiquidity += newTokensForLiquidity;
583   }
584
585
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 582

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
581   tokensForMarketing += newTokensForMarketing;
582   tokensForLiquidity += newTokensForLiquidity;
583   }
584
585   if (fees > 0) {
586
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 587

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
586   super._transfer(from, address(this), fees);
587   amount -= fees;
588   }
589   }
590
591
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 623

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
622   function swapBack() internal {
623   if (tokensForLiquidity + tokensForDev + tokensForMarketing == 0) {
624   return;
625   }
626
627
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 623

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
622   function swapBack() internal {
623   if (tokensForLiquidity + tokensForDev + tokensForMarketing == 0) {
624   return;
625   }
626
627
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 627

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
626
627   uint256 liquidity = tokensForLiquidity / 2;
628   uint256 amountToSwapForETH = tokensForDev + tokensForMarketing +
(tokensForLiquidity - liquidity);
629   swapTokensForEth(amountToSwapForETH);
630
631
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 628

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
627   uint256 liquidity = tokensForLiquidity / 2;
628   uint256 amountToSwapForETH = tokensForDev + tokensForMarketing +
(tokensForLiquidity - liquidity);
629   swapTokensForEth(amountToSwapForETH);
630
631   uint256 ethForLiquidity = address(this).balance * (tokensForLiquidity - liquidity)
/ amountToSwapForETH;
632
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 628

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
627   uint256 liquidity = tokensForLiquidity / 2;
628   uint256 amountToSwapForETH = tokensForDev + tokensForMarketing +
(tokensForLiquidity - liquidity);
629   swapTokensForEth(amountToSwapForETH);
630
631   uint256 ethForLiquidity = address(this).balance * (tokensForLiquidity - liquidity)
/ amountToSwapForETH;
632
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 628

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
627   uint256 liquidity = tokensForLiquidity / 2;
628   uint256 amountToSwapForETH = tokensForDev + tokensForMarketing +
(tokensForLiquidity - liquidity);
629   swapTokensForEth(amountToSwapForETH);
630
631   uint256 ethForLiquidity = address(this).balance * (tokensForLiquidity - liquidity)
/ amountToSwapForETH;
632
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 631

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
630
631   uint256 ethForLiquidity = address(this).balance * (tokensForLiquidity - liquidity)
/ amountToSwapForETH;
632
633   if (liquidity > 0 && ethForLiquidity > 0) {
634   _addLiquidity(liquidity, ethForLiquidity);
635
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 631

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
630
631   uint256 ethForLiquidity = address(this).balance * (tokensForLiquidity - liquidity)
/ amountToSwapForETH;
632
633   if (liquidity > 0 && ethForLiquidity > 0) {
634   _addLiquidity(liquidity, ethForLiquidity);
635
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 631

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
630
631   uint256 ethForLiquidity = address(this).balance * (tokensForLiquidity - liquidity)
/ amountToSwapForETH;
632
633   if (liquidity > 0 && ethForLiquidity > 0) {
634   _addLiquidity(liquidity, ethForLiquidity);
635
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 637

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
636
637   if (tokensForMarketing + tokensForDev > 0) {
638   uint256 remainingBalance = address(this).balance;
639   uint256 amountForMarketing = remainingBalance * tokensForMarketing /
(tokensForMarketing + tokensForDev);
640   uint256 amountForDev = remainingBalance - amountForMarketing;
641
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 639

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
638   uint256 remainingBalance = address(this).balance;
639   uint256 amountForMarketing = remainingBalance * tokensForMarketing /
(tokensForMarketing + tokensForDev);
640   uint256 amountForDev = remainingBalance - amountForMarketing;
641
642   if (amountForMarketing > 0) {
643
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 639

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
638   uint256 remainingBalance = address(this).balance;
639   uint256 amountForMarketing = remainingBalance * tokensForMarketing /
(tokensForMarketing + tokensForDev);
640   uint256 amountForDev = remainingBalance - amountForMarketing;
641
642   if (amountForMarketing > 0) {
643
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 639

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
638   uint256 remainingBalance = address(this).balance;
639   uint256 amountForMarketing = remainingBalance * tokensForMarketing /
(tokensForMarketing + tokensForDev);
640   uint256 amountForDev = remainingBalance - amountForMarketing;
641
642   if (amountForMarketing > 0) {
643
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 640

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
639   uint256 amountForMarketing = remainingBalance * tokensForMarketing /
(tokensForMarketing + tokensForDev);
640   uint256 amountForDev = remainingBalance - amountForMarketing;
641
642   if (amountForMarketing > 0) {
643   marketingReceiver.call{value: amountForMarketing}("");
644
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

low SEVERITY

The current pragma Solidity directive is ""^0.8.17"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

Source File

- KingdomToken.sol

Locations

```
5   // SPDX-License-Identifier: MIT
6   pragma solidity ^0.8.17;
7
8   /*
9
10
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 296

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
295   // a value of 50 => 5% max. A value of 150 => 15% max
296   bool maxSellFeeSet = false;
297   bool maxBuyFeeSet = false;
298   uint256 maxSellFee;
299   uint256 maxBuyFee;
300
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 297

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
296   bool maxSellFeeSet = false;
297   bool maxBuyFeeSet = false;
298   uint256 maxSellFee;
299   uint256 maxBuyFee;
300
301
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 298

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
297   bool maxBuyFeeSet = false;
298   uint256 maxSellFee;
299   uint256 maxBuyFee;
300
301   bool public airdropComplete = false;
302
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 372

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
371   for (uint i=0; i<holders.length; i++) {
372   super._transfer(address(this), holders[i], amounts[i]);
373   airdropAmount[holders[i]] += amounts[i];
374   }
375   }
376
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 372

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
371   for (uint i=0; i<holders.length; i++) {
372   super._transfer(address(this), holders[i], amounts[i]);
373   airdropAmount[holders[i]] += amounts[i];
374   }
375   }
376
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 373

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
372   super._transfer(address(this), holders[i], amounts[i]);
373   airdropAmount[holders[i]] += amounts[i];
374   }
375   }
376
377
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 373

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
372   super._transfer(address(this), holders[i], amounts[i]);
373   airdropAmount[holders[i]] += amounts[i];
374   }
375   }
376
377
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 610

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
609   address[] memory path = new address[](2);
610   path[0] = address(this);
611   path[1] = router.WETH();
612   _approve(address(this), address(router), tokenAmount);
613   router.swapExactTokensForETHSupportingFeeOnTransferTokens(
614
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 611

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
610   path[0] = address(this);
611   path[1] = router.WETH();
612   _approve(address(this), address(router), tokenAmount);
613   router.swapExactTokensForETHSupportingFeeOnTransferTokens(
614   tokenAmount,
615
```

SWC-104 | UNCHECKED RETURN VALUE FROM LOW-LEVEL EXTERNAL CALL.

LINE 643

medium SEVERITY

Low-level external calls return a boolean value. If the callee halts with an exception, 'false' is returned and execution continues in the caller. The caller should check whether an exception happened and react accordingly to avoid unexpected behavior. For example it is often desirable to wrap low-level external calls in require() so the transaction is reverted if the call fails.

Source File

- KingdomToken.sol

Locations

```
642   if (amountForMarketing > 0) {
643   marketingReceiver.call{value: amountForMarketing}("");
644   }
645
646   if (amountForDev > 0) {
647
```

SWC-104 | UNCHECKED RETURN VALUE FROM LOW-LEVEL EXTERNAL CALL.

LINE 647

medium SEVERITY

Low-level external calls return a boolean value. If the callee halts with an exception, 'false' is returned and execution continues in the caller. The caller should check whether an exception happened and react accordingly to avoid unexpected behavior. For example it is often desirable to wrap low-level external calls in require() so the transaction is reverted if the call fails.

Source File

- KingdomToken.sol

Locations

```
646   if (amountForDev > 0) {
647   devReceiver.call{value: amountForDev}("");
648   }
649   }
650
651
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 299

low SEVERITY

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

Source File

- KingdomToken.sol

Locations

```
298   uint256 maxSellFee;
299   uint256 maxBuyFee;
300
301   bool public airdropComplete = false;
302   bool public vestingFinished = false;
303
```

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.