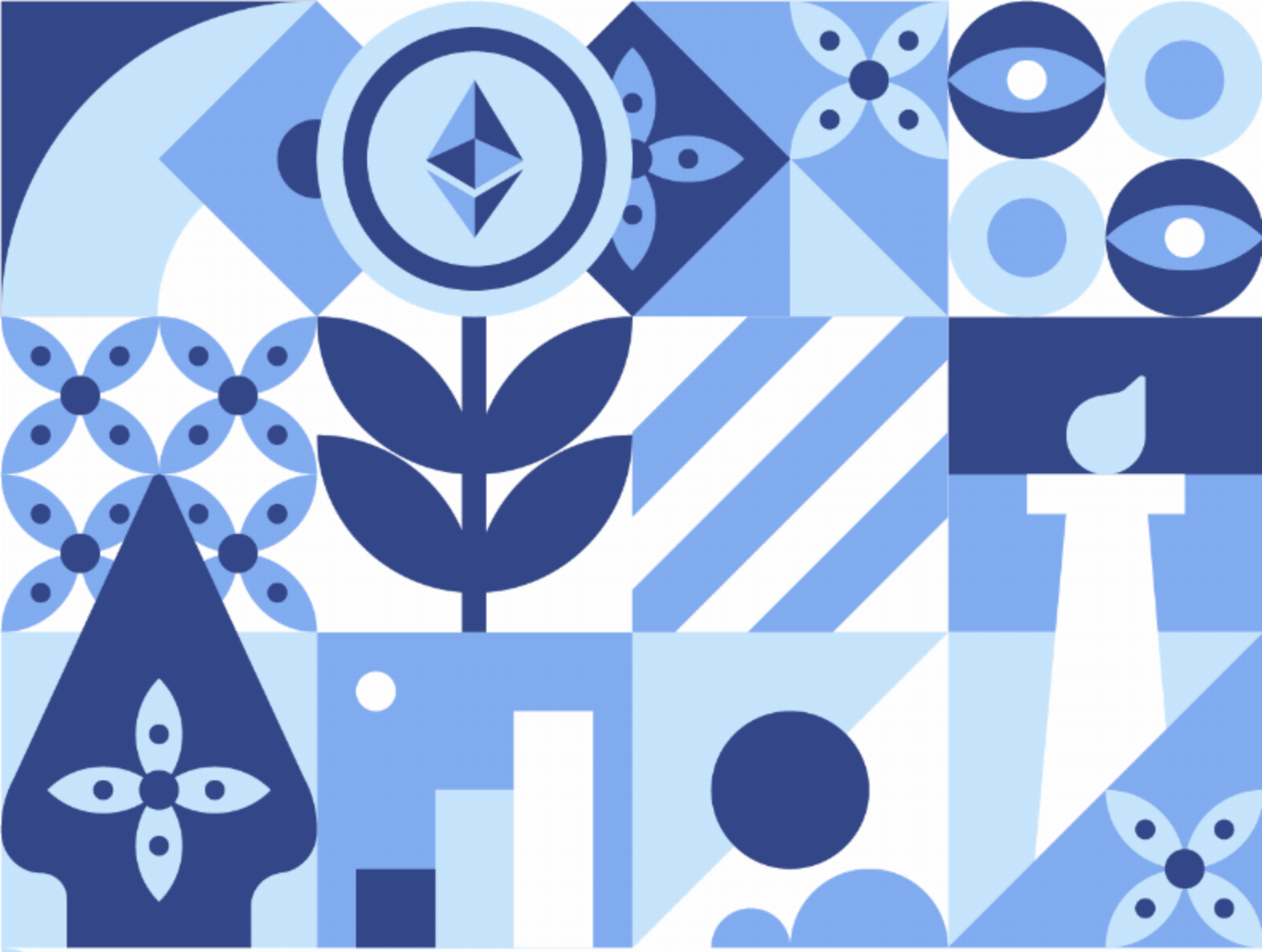

Golden

Smart Contract Audit Report

SYSFIXED

26 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
Golden	Golden	Ethereum

Addresses

Contract address	0x40f8a48d7579039d42ff832e67fad7d891640bdb
Contract deployer address	0x3199Aa75b842d843BCc079703EBff6cec61AD8c1

Project Website

https://goldensmartcontract.com/

Codebase

https://etherscan.io/address/0x40f8a48d7579039d42ff832e67fad7d891640bdb#code

SUMMARY

$golden operates in a zero-fee environment, making it an attractive option for those looking to participate in the DeFi space without incurring high transaction costs.

Contract Summary

Documentation Quality

Golden 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 Golden 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 390, 409, 431, 464, 466, 487, 488, 513, 515, 611, 625, 640, 641, 654, 666, 681, 695, 709, 723, 739, 762, 785, 811, 1145, 1147, 1148, 1149, 1149, 1154, 1154, 1159, 1159, 1208, 1208, 1212, 1212, 1221, 1221, 1221, 1224, 1224, 1229, 1229, 1229, 1232, 1232, 1255, 1255, 1267, 1267, 1367, 1382, 1412, 1431, 1431, 1431, 1432, 1432, 1432, 1433, 1433, 1433, 1438, 1438, 1438, 1439, 1439, 1439, 1440, 1440, 1440, 1447, 1488, 1488, 1497, 1498, 1502, 1502, 1502, 1518, 1518 and 1587.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on lines 6.
- SWC-110 SWC-123 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 1456 and 1457.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 1349 and 1353.
- SWC-120 | It is recommended to use external sources of randomness via oracles on lines 1350 and 1353.

CONCLUSION

We have audited the Golden project released on January 2023 to discover issues and identify potential security vulnerabilities in Golden 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 Golden 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. We recommend avoiding "tx.origin" issue 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. We recommend to don't using any of those environment variables as sources of randomness and be aware that the use of these variables introduces a certain level of trust in miners.

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	Wednesday Jan 25 2023 03:19:33 GMT+0000 (Coordinated Universal Time)
Finished	Thursday Jan 26 2023 17:02:47 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	Golden.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 390

low SEVERITY

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

Source File

- Golden.sol

Locations

```
389   unchecked {
390   _approve(sender, _msgSender(), currentAllowance - amount);
391   }
392
393   return true;
394
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 409

low SEVERITY

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

Source File

- Golden.sol

Locations

```
408   function increaseAllowance(address spender, uint256 addedValue) public virtual
returns (bool) {
409   _approve(_msgSender(), spender, _allowances[_msgSender()][spender] + addedValue);
410   return true;
411   }
412
413
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 431

low SEVERITY

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

Source File

- Golden.sol

Locations

```
430   unchecked {
431   _approve(_msgSender(), spender, currentAllowance - subtractedValue);
432   }
433
434   return true;
435
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 464

low SEVERITY

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

Source File

- Golden.sol

Locations

```
463   unchecked {
464   _balances[sender] = senderBalance - amount;
465   }
466   _balances[recipient] += amount;
467
468
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 466

low SEVERITY

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

Source File

- Golden.sol

Locations

```
465   }
466   _balances[recipient] += amount;
467
468   emit Transfer(sender, recipient, amount);
469
470
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 487

low SEVERITY

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

Source File

- Golden.sol

Locations

```
486
487   _totalSupply += amount;
488   _balances[account] += amount;
489   emit Transfer(address(0), account, amount);
490
491
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 488

low SEVERITY

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

Source File

- Golden.sol

Locations

```
487   _totalSupply += amount;
488   _balances[account] += amount;
489   emit Transfer(address(0), account, amount);
490
491   _afterTokenTransfer(address(0), account, amount);
492
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 513

low SEVERITY

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

Source File

- Golden.sol

Locations

```
512   unchecked {
513   _balances[account] = accountBalance - amount;
514   }
515   _totalSupply -= amount;
516
517
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 515

low SEVERITY

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

Source File

- Golden.sol

Locations

```
514   }
515   _totalSupply -= amount;
516
517   emit Transfer(account, address(0), amount);
518
519
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 611

low SEVERITY

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

Source File

- Golden.sol

Locations

```
610   unchecked {
611   uint256 c = a + b;
612   if (c < a) return (false, 0);
613   return (true, c);
614   }
615
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 625

low SEVERITY

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

Source File

- Golden.sol

Locations

```
624   if (b > a) return (false, 0);
625   return (true, a - b);
626   }
627   }
628
629
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 640

low SEVERITY

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

Source File

- Golden.sol

Locations

```
639   if (a == 0) return (true, 0);
640   uint256 c = a * b;
641   if (c / a != b) return (false, 0);
642   return (true, c);
643   }
644
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 641

low SEVERITY

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

Source File

- Golden.sol

Locations

```
640   uint256 c = a * b;
641   if (c / a != b) return (false, 0);
642   return (true, c);
643   }
644   }
645
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 654

low SEVERITY

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

Source File

- Golden.sol

Locations

```
653   if (b == 0) return (false, 0);
654   return (true, a / b);
655   }
656   }
657
658
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 666

low SEVERITY

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

Source File

- Golden.sol

Locations

```
665   if (b == 0) return (false, 0);
666   return (true, a % b);
667   }
668   }
669
670
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 681

low SEVERITY

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

Source File

- Golden.sol

Locations

```
680   function add(uint256 a, uint256 b) internal pure returns (uint256) {
681   return a + b;
682   }
683
684   /**
685
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 695

low SEVERITY

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

Source File

- Golden.sol

Locations

```
694   function sub(uint256 a, uint256 b) internal pure returns (uint256) {
695   return a - b;
696   }
697
698   /**
699
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 709

low SEVERITY

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

Source File

- Golden.sol

Locations

```
708   function mul(uint256 a, uint256 b) internal pure returns (uint256) {
709   return a * b;
710   }
711
712   /**
713
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- Golden.sol

Locations

```
722   function div(uint256 a, uint256 b) internal pure returns (uint256) {
723   return a / b;
724   }
725
726   /**
727
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 739

low SEVERITY

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

Source File

- Golden.sol

Locations

```
738   function mod(uint256 a, uint256 b) internal pure returns (uint256) {
739   return a % b;
740   }
741
742   /**
743
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 762

low SEVERITY

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

Source File

- Golden.sol

Locations

```
761   require(b <= a, errorMessage);
762   return a - b;
763   }
764   }
765
766
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 785

low SEVERITY

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

Source File

- Golden.sol

Locations

```
784   require(b > 0, errorMessage);
785   return a / b;
786   }
787   }
788
789
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 811

low SEVERITY

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

Source File

- Golden.sol

Locations

```
810   require(b > 0, errorMessage);
811   return a % b;
812   }
813   }
814   }
815
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1145

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1144
1145   uint256 totalSupply = 1_000_000_000 * 1e18;
1146
1147   maxTransactionAmount =  1_000_000_000 * 1e18; // 100% from total supply
maxTransactionAmountTxn
1148   maxWallet =  1_000_000_000 * 1e18; // 100% from total supply maxWallet
1149
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1147

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1146
1147   maxTransactionAmount =  1_000_000_000 * 1e18; // 100% from total supply
maxTransactionAmountTxn
1148   maxWallet =  1_000_000_000 * 1e18; // 100% from total supply maxWallet
1149   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
1150
1151
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1148

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1147   maxTransactionAmount =  1_000_000_000 * 1e18; // 100% from total supply
maxTransactionAmountTxn
1148   maxWallet =  1_000_000_000 * 1e18; // 100% from total supply maxWallet
1149   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
1150
1151   buyMarketingFee = _buyMarketingFee;
1152
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1149

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1148   maxWallet =  1_000_000_000 * 1e18; // 100% from total supply maxWallet
1149   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
1150
1151   buyMarketingFee = _buyMarketingFee;
1152   buyLiquidityFee = _buyLiquidityFee;
1153
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1149

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1148   maxWallet =  1_000_000_000 * 1e18; // 100% from total supply maxWallet
1149   swapTokensAtAmount = (totalSupply * 5) / 10000; // 0.05% swap wallet
1150
1151   buyMarketingFee = _buyMarketingFee;
1152   buyLiquidityFee = _buyLiquidityFee;
1153
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1154

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1153   buyDevFee = _buyDevFee;
1154   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1155
1156   sellMarketingFee = _sellMarketingFee;
1157   sellLiquidityFee = _sellLiquidityFee;
1158
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1154

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1153   buyDevFee = _buyDevFee;
1154   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1155
1156   sellMarketingFee = _sellMarketingFee;
1157   sellLiquidityFee = _sellLiquidityFee;
1158
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1159

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1158   sellDevFee = _sellDevFee;
1159   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1160
1161   marketingWallet = address(0x000000000000000000000000000000000000dEaD); //
1162   devWallet = address(0x000000000000000000000000000000000000dEaD); //
1163
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1159

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1158   sellDevFee = _sellDevFee;
1159   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1160
1161   marketingWallet = address(0x000000000000000000000000000000000000dEaD); //
1162   devWallet = address(0x000000000000000000000000000000000000dEaD); //
1163
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1208

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1207   require(
1208   newAmount >= (totalSupply() * 1) / 100000,
1209   "Swap amount cannot be lower than 0.001% total supply."
1210   );
1211   require(
1212
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1208

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1207   require(
1208   newAmount >= (totalSupply() * 1) / 100000,
1209   "Swap amount cannot be lower than 0.001% total supply."
1210   );
1211   require(
1212
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1212

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1211   require(
1212   newAmount <= (totalSupply() * 5) / 1000,
1213   "Swap amount cannot be higher than 0.5% total supply."
1214   );
1215   swapTokensAtAmount = newAmount;
1216
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1212

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1211   require(
1212   newAmount <= (totalSupply() * 5) / 1000,
1213   "Swap amount cannot be higher than 0.5% total supply."
1214   );
1215   swapTokensAtAmount = newAmount;
1216
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1221

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1220   require(
1221   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1222   "Cannot set maxTransactionAmount lower than 0.1%"
1223   );
1224   maxTransactionAmount = newNum * (10**18);
1225
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1221

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1220   require(
1221   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1222   "Cannot set maxTransactionAmount lower than 0.1%"
1223   );
1224   maxTransactionAmount = newNum * (10**18);
1225
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1221

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1220   require(
1221   newNum >= ((totalSupply() * 1) / 1000) / 1e18,
1222   "Cannot set maxTransactionAmount lower than 0.1%"
1223   );
1224   maxTransactionAmount = newNum * (10**18);
1225
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1224

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1223   );
1224   maxTransactionAmount = newNum * (10**18);
1225   }
1226
1227   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1228
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1224

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1223   );
1224   maxTransactionAmount = newNum * (10**18);
1225   }
1226
1227   function updateMaxWalletAmount(uint256 newNum) external onlyOwner {
1228
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1228   require(
1229   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1230   "Cannot set maxWallet lower than 0.5%"
1231   );
1232   maxWallet = newNum * (10**18);
1233
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1228   require(
1229   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1230   "Cannot set maxWallet lower than 0.5%"
1231   );
1232   maxWallet = newNum * (10**18);
1233
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1229

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1228   require(
1229   newNum >= ((totalSupply() * 5) / 1000) / 1e18,
1230   "Cannot set maxWallet lower than 0.5%"
1231   );
1232   maxWallet = newNum * (10**18);
1233
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1232

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1231   );
1232   maxWallet = newNum * (10**18);
1233   }
1234
1235   function excludeFromMaxTransaction(address updAds, bool isEx)
1236
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 1232

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1231   );
1232   maxWallet = newNum * (10**18);
1233   }
1234
1235   function excludeFromMaxTransaction(address updAds, bool isEx)
1236
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1255

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1254   buyDevFee = _devFee;
1255   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1256   require(buyTotalFees <= 11, "Must keep fees at 11% or less");
1257   }
1258
1259
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1255

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1254   buyDevFee = _devFee;
1255   buyTotalFees = buyMarketingFee + buyLiquidityFee + buyDevFee;
1256   require(buyTotalFees <= 11, "Must keep fees at 11% or less");
1257   }
1258
1259
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1267

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1266   sellDevFee = _devFee;
1267   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1268   require(sellTotalFees <= 11, "Must keep fees at 11% or less");
1269   }
1270
1271
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1267

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1266   sellDevFee = _devFee;
1267   sellTotalFees = sellMarketingFee + sellLiquidityFee + sellDevFee;
1268   require(sellTotalFees <= 11, "Must keep fees at 11% or less");
1269   }
1270
1271
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1367

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1366   require(
1367   amount + balanceOf(to) <= maxWallet,
1368   "Max wallet exceeded"
1369   );
1370   }
1371
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1382

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1381   require(
1382   amount + balanceOf(to) <= maxWallet,
1383   "Max wallet exceeded"
1384   );
1385   }
1386
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1412

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1411   lpBurnEnabled &&
1412   block.timestamp >= lastLpBurnTime + lpBurnFrequency &&
1413   !_isExcludedFromFees[from]
1414   ) {
1415   autoBurnLiquidityPairTokens();
1416
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1431

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1430   fees = amount.mul(sellTotalFees).div(100);
1431   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1431

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1430   fees = amount.mul(sellTotalFees).div(100);
1431   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1431

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1430   fees = amount.mul(sellTotalFees).div(100);
1431   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1432

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1431   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435   // on buy
1436
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1432

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1431   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435   // on buy
1436
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1432

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1431   tokensForLiquidity += (fees * sellLiquidityFee) / sellTotalFees;
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435   // on buy
1436
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1433

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435   // on buy
1436   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1437
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1433

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435   // on buy
1436   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1437
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1433

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1432   tokensForDev += (fees * sellDevFee) / sellTotalFees;
1433   tokensForMarketing += (fees * sellMarketingFee) / sellTotalFees;
1434   }
1435   // on buy
1436   else if (automatedMarketMakerPairs[from] && buyTotalFees > 0) {
1437
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1437   fees = amount.mul(buyTotalFees).div(100);
1438   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1437   fees = amount.mul(buyTotalFees).div(100);
1438   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1438

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1437   fees = amount.mul(buyTotalFees).div(100);
1438   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1439

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1438   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
1443
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1439

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1438   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
1443
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1439

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1438   tokensForLiquidity += (fees * buyLiquidityFee) / buyTotalFees;
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
1443
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 1440

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
1443   if (fees > 0) {
1444
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1440

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
1443   if (fees > 0) {
1444
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1440

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1439   tokensForDev += (fees * buyDevFee) / buyTotalFees;
1440   tokensForMarketing += (fees * buyMarketingFee) / buyTotalFees;
1441   }
1442
1443   if (fees > 0) {
1444
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 1447

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1446
1447   amount -= fees;
1448   }
1449
1450   super._transfer(from, to, amount);
1451
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1488

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1487   uint256 contractBalance = balanceOf(address(this));
1488   uint256 totalTokensToSwap = tokensForLiquidity +
1489   tokensForMarketing +
1490   tokensForDev;
1491   bool success;
1492
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1488

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1487   uint256 contractBalance = balanceOf(address(this));
1488   uint256 totalTokensToSwap = tokensForLiquidity +
1489   tokensForMarketing +
1490   tokensForDev;
1491   bool success;
1492
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1497

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1496
1497   if (contractBalance > swapTokensAtAmount * 20) {
1498   contractBalance = swapTokensAtAmount * 20;
1499   }
1500
1501
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1498

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1497   if (contractBalance > swapTokensAtAmount * 20) {
1498   contractBalance = swapTokensAtAmount * 20;
1499   }
1500
1501   // Halve the amount of liquidity tokens
1502
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1502

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1501   // Halve the amount of liquidity tokens
1502   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1503   totalTokensToSwap /
1504   2;
1505   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1506
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 1502

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1501   // Halve the amount of liquidity tokens
1502   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1503   totalTokensToSwap /
1504   2;
1505   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1506
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 1502

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1501   // Halve the amount of liquidity tokens
1502   uint256 liquidityTokens = (contractBalance * tokensForLiquidity) /
1503   totalTokensToSwap /
1504   2;
1505   uint256 amountToSwapForETH = contractBalance.sub(liquidityTokens);
1506
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1518

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1517
1518   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1519
1520   tokensForLiquidity = 0;
1521   tokensForMarketing = 0;
1522
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 1518

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1517
1518   uint256 ethForLiquidity = ethBalance - ethForMarketing - ethForDev;
1519
1520   tokensForLiquidity = 0;
1521   tokensForMarketing = 0;
1522
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 1587

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1586   require(
1587   block.timestamp > lastManualLpBurnTime + manualBurnFrequency,
1588   "Must wait for cooldown to finish"
1589   );
1590   require(percent <= 1000, "May not nuke more than 10% of tokens in LP");
1591
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

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

- Golden.sol

Locations

```
5   // SPDX-License-Identifier: MIT
6   pragma solidity =0.8.10 >=0.8.10 >=0.8.0 <0.9.0;
7   pragma experimental ABIEncoderV2;
8
9   // GOLDEN Smart Contract
10
```

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

LINE 1349

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

- Golden.sol

Locations

```
1348   require(
1349   _holderLastTransferTimestamp[tx.origin] <
1350   block.number,
1351   "_transfer:: Transfer Delay enabled.  Only one purchase per block allowed."
1352   );
1353
```

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

LINE 1353

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

- Golden.sol

Locations

```
1352   );
1353   _holderLastTransferTimestamp[tx.origin] = block.number;
1354   }
1355   }
1356
1357
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1456

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1455   address[] memory path = new address[](2);
1456   path[0] = address(this);
1457   path[1] = uniswapV2Router.WETH();
1458
1459   _approve(address(this), address(uniswapV2Router), tokenAmount);
1460
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 1457

low SEVERITY

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

Source File

- Golden.sol

Locations

```
1456   path[0] = address(this);
1457   path[1] = uniswapV2Router.WETH();
1458
1459   _approve(address(this), address(uniswapV2Router), tokenAmount);
1460
1461
```

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

LINE 1350

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

- Golden.sol

Locations

```
1349   _holderLastTransferTimestamp[tx.origin] <
1350   block.number,
1351   "_transfer:: Transfer Delay enabled.  Only one purchase per block allowed."
1352   );
1353   _holderLastTransferTimestamp[tx.origin] = block.number;
1354
```

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

LINE 1353

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

- Golden.sol

Locations

```
1352   );
1353   _holderLastTransferTimestamp[tx.origin] = block.number;
1354   }
1355   }
1356
1357
```

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.