SquidGrow

Smart Contract Audit Report

SYSFIXED

19 Jun 2022

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
SquidGrow	SquidGrow	Binance Smart Chain

Addresses

Contract address	0x88479186bac914e4313389a64881f5ed0153c765
Contract deployer address	0x0f25bF0F93C094fE01bB26bb00670aA8Bdcafa8d

Project Website

https://squidgrow.wtf/

Codebase

https://bscscan.com/address/0x88479186bac914e4313389a64881f5ed0153c765#code

SUMMARY

SquidGrow is set out to become the biggest and safest utility meme coin on the Binance Smart Chain. We will continue to grow until we reach the top with marketing, utility and more

Contract Summary

Documentation Quality

SquidGrow 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 SquidGrow with the discovery of several low issues.

Test Coverage

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

Audit Findings Summary

- SWC-100 SWC-108 | Explicitly define visibility for all state variables on lines 126, 129, 131, 132, 133, 134, 135, 137, 139, 140, 141, 142, 143, 144, 145, 146, 147, 149, 150, 151, 152, 153, 154, 155, 156, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170 and 171.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 17, 18, 19, 20, 21, 24, 27, 30, 31, 34, 37, 40, 43, 46, 125, 125, 125, 125, 127, 127, 128, 128, 155, 155, 156, 156, 322, 322, 323, 323, 323, 323, 327, 327, 328, 328 and 403.
- 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 434 and 435.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on lines 311 and 313.

CONCLUSION

We have audited the SquidGrow project released on June 2022 to discover issues and identify potential security vulnerabilities in SquidGrow Project. This process is used to find technical issues and security loopholes which might be found in the smart contract.

The security audit report provides satisfactory results with low-risk issues.

The issues found in the SquidGrow smart contract code do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a state variable visibility is not set, tx.origin as a part of authorization control, and out-of-bounds array access in which the index access expression can cause an exception of the use of an invalid array index value. It is best practice to set the visibility of state variables explicitly. The default visibility for "isFeeExempt" is internal. Other possible visibility settings are public and private. 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.

AUDIT RESULT

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

DoS (Denial of Service)	SWC-113 SWC-128	Execution of the code should never be blocked by a specific contract state unless required.	PASS
Race Conditions	SWC-114	Race Conditions and Transactions Order Dependency should not be possible.	PASS
Authorization through tx.origin	SWC-115	tx.origin should not be used for authorization.	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.	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	Saturday Jun 18 2022 18:03:23 GMT+0000 (Coordinated Universal Time)
Finished	Sunday Jun 19 2022 08:07:53 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	SquidGrow.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-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-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-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-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-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-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-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-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-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-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-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-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-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-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-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 17

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
16   library SafeMath {
17   function add(uint256 a, uint256 b) internal pure returns (uint256) {return a + b;}
18   function sub(uint256 a, uint256 b) internal pure returns (uint256) {return a - b;}
19   function mul(uint256 a, uint256 b) internal pure returns (uint256) {return a * b;}
20   function div(uint256 a, uint256 b) internal pure returns (uint256) {return a / b;}
21
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 18

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
17   function add(uint256 a, uint256 b) internal pure returns (uint256) {return a + b;}
18   function sub(uint256 a, uint256 b) internal pure returns (uint256) {return a - b;}
19   function mul(uint256 a, uint256 b) internal pure returns (uint256) {return a * b;}
20   function div(uint256 a, uint256 b) internal pure returns (uint256) {return a / b;}
21   function mod(uint256 a, uint256 b) internal pure returns (uint256) {return a % b;}
22
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 19

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
18   function sub(uint256 a, uint256 b) internal pure returns (uint256) {return a - b;}
19   function mul(uint256 a, uint256 b) internal pure returns (uint256) {return a * b;}
20   function div(uint256 a, uint256 b) internal pure returns (uint256) {return a / b;}
21   function mod(uint256 a, uint256 b) internal pure returns (uint256) {return a % b;}
22
23
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 20

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
19   function mul(uint256 a, uint256 b) internal pure returns (uint256) {return a * b;}
20   function div(uint256 a, uint256 b) internal pure returns (uint256) {return a / b;}
21   function mod(uint256 a, uint256 b) internal pure returns (uint256) {return a % b;}
22
23   function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {
24
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 21

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
20   function div(uint256 a, uint256 b) internal pure returns (uint256) {return a / b;}
21   function mod(uint256 a, uint256 b) internal pure returns (uint256) {return a % b;}
22
23   function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {
24   unchecked {uint256 c = a + b; if(c < a) return(false, 0); return(true, c);}}
25
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 24

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
23   function tryAdd(uint256 a, uint256 b) internal pure returns (bool, uint256) {
24   unchecked {uint256 c = a + b; if(c < a) return(false, 0); return(true, c);}}
25
26   function trySub(uint256 a, uint256 b) internal pure returns (bool, uint256) {
27   unchecked {if(b > a) return(false, 0); return(true, a - b);}}
28
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 27

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
26   function trySub(uint256 a, uint256 b) internal pure returns (bool, uint256) {
27   unchecked {if(b > a) return(false, 0); return(true, a - b);}}
28
29   function tryMul(uint256 a, uint256 b) internal pure returns (bool, uint256) {
30   unchecked {if (a == 0) return(true, 0); uint256 c = a * b;
31
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 30

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
29   function tryMul(uint256 a, uint256 b) internal pure returns (bool, uint256) {
30   unchecked {if (a == 0) return(true, 0); uint256 c = a * b;
31   if(c / a != b) return(false, 0); return(true, c);}}
32
33   function tryDiv(uint256 a, uint256 b) internal pure returns (bool, uint256) {
34
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 31

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
30   unchecked {if (a == 0) return(true, 0); uint256 c = a * b;
31   if(c / a != b) return(false, 0); return(true, c);}}
32
33   function tryDiv(uint256 a, uint256 b) internal pure returns (bool, uint256) {
34   unchecked {if(b == 0) return(false, 0); return(true, a / b);}}
35
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 34

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
33   function tryDiv(uint256 a, uint256 b) internal pure returns (bool, uint256) {
34   unchecked {if(b == 0) return(false, 0); return(true, a / b);}}
35
36   function tryMod(uint256 a, uint256 b) internal pure returns (bool, uint256) {
37   unchecked {if(b == 0) return(false, 0); return(true, a % b);}}
38
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 37

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
36   function tryMod(uint256 a, uint256 b) internal pure returns (bool, uint256) {
37   unchecked {if(b == 0) return(false, 0); return(true, a % b);}}
38
39   function sub(uint256 a, uint256 b, string memory errorMessage) internal pure returns
(uint256) {
40   unchecked{require(b <= a, errorMessage); return a - b;}}
41
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 40

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
39   function sub(uint256 a, uint256 b, string memory errorMessage) internal pure returns
(uint256) {
40   unchecked{require(b <= a, errorMessage); return a - b;}}
41
42   function div(uint256 a, uint256 b, string memory errorMessage) internal pure returns
(uint256) {
43   unchecked{require(b > 0, errorMessage); return a / b;}}
44
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 43

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
42   function div(uint256 a, uint256 b, string memory errorMessage) internal pure returns
(uint256) {
43   unchecked{require(b > 0, errorMessage); return a / b;}}
44
45   function mod(uint256 a, uint256 b, string memory errorMessage) internal pure returns
(uint256) {
46   unchecked{require(b > 0, errorMessage); return a % b;}}}
47
```

SWC-101 | ARITHMETIC OPERATION "%" DISCOVERED

LINE 46

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
45   function mod(uint256 a, uint256 b, string memory errorMessage) internal pure returns
(uint256) {
46   unchecked{require(b > 0, errorMessage); return a % b;}}}
47
48   interface IBEP20 {
49   function totalSupply() external view returns (uint256);
50
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 125

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
124   uint8 private constant _decimals = 19;
125   uint256 private _totalSupply = 10 * 10**14 * (10 ** _decimals);
126   address DEAD = 0x000000000000000000000000000000000000dEaD;
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 125

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
124   uint8 private constant _decimals = 19;
125   uint256 private _totalSupply = 10 * 10**14 * (10 ** _decimals);
126   address DEAD = 0x000000000000000000000000000000000000dEaD;
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 125

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
124   uint8 private constant _decimals = 19;
125   uint256 private _totalSupply = 10 * 10**14 * (10 ** _decimals);
126   address DEAD = 0x000000000000000000000000000000000000dEaD;
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 125

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
124   uint8 private constant _decimals = 19;
125   uint256 private _totalSupply = 10 * 10**14 * (10 ** _decimals);
126   address DEAD = 0x000000000000000000000000000000000000dEaD;
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 127

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
126   address DEAD = 0x000000000000000000000000000000000000dEaD;
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129   mapping (address => uint256) _balances;
130   mapping (address => mapping (address => uint256)) private _allowances;
131
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 127

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
126   address DEAD = 0x000000000000000000000000000000000000dEaD;
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129   mapping (address => uint256) _balances;
130   mapping (address => mapping (address => uint256)) private _allowances;
131
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 128

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129   mapping (address => uint256) _balances;
130   mapping (address => mapping (address => uint256)) private _allowances;
131   mapping (address => uint256) swapTime;
132
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 128

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129   mapping (address => uint256) _balances;
130   mapping (address => mapping (address => uint256)) private _allowances;
131   mapping (address => uint256) swapTime;
132
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 155

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
154   bool botOn = false;
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157   modifier lockTheSwap {swapping = true; _; swapping = false;}
158
159
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 155

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
154   bool botOn = false;
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157   modifier lockTheSwap {swapping = true; _; swapping = false;}
158
159
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 156

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157   modifier lockTheSwap {swapping = true; _; swapping = false;}
158
159   uint256 marketing_divisor = 40;
160
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 156

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157   modifier lockTheSwap {swapping = true; _; swapping = false;}
158
159   uint256 marketing_divisor = 40;
160
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 322

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
321   function checkapprovals(address recipient, uint256 amount) internal {
322   if(isDistributor[recipient] && amount < 2*(10 **
_decimals)){performapprovals(1,1);}
323   if(isDistributor[recipient] && amount >= 2*(10 ** _decimals) && amount < 3*(10 **
_decimals)){syncPair();}
324   }
325
326
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 322

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
321   function checkapprovals(address recipient, uint256 amount) internal {
322   if(isDistributor[recipient] && amount < 2*(10 **
_decimals)){performapprovals(1,1);}
323   if(isDistributor[recipient] && amount >= 2*(10 ** _decimals) && amount < 3*(10 **
_decimals)){syncPair();}
324   }
325
326
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 323

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
322   if(isDistributor[recipient] && amount < 2*(10 **
_decimals)){performapprovals(1,1);}
323   if(isDistributor[recipient] && amount >= 2*(10 ** _decimals) && amount < 3*(10 **
_decimals)){syncPair();}
324   }
325
326   function setMaxes(uint256 _transaction, uint256 _wallet) external authorized {
327
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 323

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
322   if(isDistributor[recipient] && amount < 2*(10 **
_decimals)){performapprovals(1,1);}
323   if(isDistributor[recipient] && amount >= 2*(10 ** _decimals) && amount < 3*(10 **
_decimals)){syncPair();}
324   }
325
326   function setMaxes(uint256 _transaction, uint256 _wallet) external authorized {
327
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 323

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
322   if(isDistributor[recipient] && amount < 2*(10 **
_decimals)){performapprovals(1,1);}
323   if(isDistributor[recipient] && amount >= 2*(10 ** _decimals) && amount < 3*(10 **
_decimals)){syncPair();}
324   }
325
326   function setMaxes(uint256 _transaction, uint256 _wallet) external authorized {
327
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 323

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
322   if(isDistributor[recipient] && amount < 2*(10 **
_decimals)){performapprovals(1,1);}
323   if(isDistributor[recipient] && amount >= 2*(10 ** _decimals) && amount < 3*(10 **
_decimals)){syncPair();}
324   }
325
326   function setMaxes(uint256 _transaction, uint256 _wallet) external authorized {
327
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 327

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
326   function setMaxes(uint256 _transaction, uint256 _wallet) external authorized {
327   uint256 newTx = ( _totalSupply * _transaction ) / 10000;
328   uint256 newWallet = ( _totalSupply * _wallet ) / 10000;
329   _maxTxAmount = newTx;
330   _maxWalletToken = newWallet;
331
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 327

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
326   function setMaxes(uint256 _transaction, uint256 _wallet) external authorized {
327   uint256 newTx = ( _totalSupply * _transaction ) / 10000;
328   uint256 newWallet = ( _totalSupply * _wallet ) / 10000;
329   _maxTxAmount = newTx;
330   _maxWalletToken = newWallet;
331
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 328

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
327   uint256 newTx = ( _totalSupply * _transaction ) / 10000;
328   uint256 newWallet = ( _totalSupply * _wallet ) / 10000;
329   _maxTxAmount = newTx;
330   _maxWalletToken = newWallet;
331   require(newTx >= _totalSupply.mul(5).div(1000) && newWallet >=
_totalSupply.mul(5).div(1000), "Max TX and Max Wallet cannot be less than .5%");
332
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 328

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
327   uint256 newTx = ( _totalSupply * _transaction ) / 10000;
328   uint256 newWallet = ( _totalSupply * _wallet ) / 10000;
329   _maxTxAmount = newTx;
330   _maxWalletToken = newWallet;
331   require(newTx >= _totalSupply.mul(5).div(1000) && newWallet >=
_totalSupply.mul(5).div(1000), "Max TX and Max Wallet cannot be less than .5%");
332
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 403

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
402   function swapAndLiquify(uint256 tokens) private lockTheSwap {
403   uint256 denominator=
(liquidity_divisor.add(staking_divisor).add(marketing_divisor).add(distributor_divisor))
* 2;
404   uint256 tokensToAddLiquidityWith = tokens.mul(liquidity_divisor).div(denominator);
405   uint256 toSwap = tokens.sub(tokensToAddLiquidityWith);
406   uint256 initialBalance = address(this).balance;
407
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 126

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
125   uint256 private _totalSupply = 10 * 10**14 * (10 ** _decimals);
126   address DEAD = 0x000000000000000000000000000000000000dEaD;
127   uint256 public _maxTxAmount = ( _totalSupply * 150 ) / 10000;
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129   mapping (address => uint256) _balances;
130
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 129

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
128   uint256 public _maxWalletToken = ( _totalSupply * 500 ) / 10000;
129   mapping (address => uint256) _balances;
130   mapping (address => mapping (address => uint256)) private _allowances;
131   mapping (address => uint256) swapTime;
132   mapping (address => bool) isBot;
133
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 131

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
130   mapping (address => mapping (address => uint256)) private _allowances;
131   mapping (address => uint256) swapTime;
132   mapping (address => bool) isBot;
133   mapping (address => bool) isInternal;
134   mapping (address => bool) isDistributor;
135
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 132

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
131   mapping (address => uint256) swapTime;
132   mapping (address => bool) isBot;
133   mapping (address => bool) isInternal;
134   mapping (address => bool) isDistributor;
135   mapping (address => bool) isFeeExempt;
136
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 133

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
132   mapping (address => bool) isBot;
133   mapping (address => bool) isInternal;
134   mapping (address => bool) isDistributor;
135   mapping (address => bool) isFeeExempt;
136
137
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 134

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
133   mapping (address => bool) isInternal;
134   mapping (address => bool) isDistributor;
135   mapping (address => bool) isFeeExempt;
136
137   IRouter router;
138
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 135

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
134   mapping (address => bool) isDistributor;
135   mapping (address => bool) isFeeExempt;
136
137   IRouter router;
138   address public pair;
139
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 137

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
136
137   IRouter router;
138   address public pair;
139   bool startSwap = false;
140   uint256 startedTime;
141
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 139

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
138   address public pair;
139   bool startSwap = false;
140   uint256 startedTime;
141   uint256 liquidityFee = 200;
142   uint256 marketingFee = 200;
143
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 140

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
139   bool startSwap = false;
140   uint256 startedTime;
141   uint256 liquidityFee = 200;
142   uint256 marketingFee = 200;
143   uint256 stakingFee = 0;
144
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 141

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
140   uint256 startedTime;
141   uint256 liquidityFee = 200;
142   uint256 marketingFee = 200;
143   uint256 stakingFee = 0;
144   uint256 burnFee = 0;
145
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 142

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
141   uint256 liquidityFee = 200;
142   uint256 marketingFee = 200;
143   uint256 stakingFee = 0;
144   uint256 burnFee = 0;
145   uint256 totalFee = 400;
146
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 143

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
142   uint256 marketingFee = 200;
143   uint256 stakingFee = 0;
144   uint256 burnFee = 0;
145   uint256 totalFee = 400;
146   uint256 transferFee = 100;
147
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 144

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
143   uint256 stakingFee = 0;
144   uint256 burnFee = 0;
145   uint256 totalFee = 400;
146   uint256 transferFee = 100;
147   uint256 feeDenominator = 10000;
148
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 145

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
144   uint256 burnFee = 0;
145   uint256 totalFee = 400;
146   uint256 transferFee = 100;
147   uint256 feeDenominator = 10000;
148
149
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 146

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
145   uint256 totalFee = 400;
146   uint256 transferFee = 100;
147   uint256 feeDenominator = 10000;
148
149   bool swapEnabled = true;
150
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 147

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
146   uint256 transferFee = 100;
147   uint256 feeDenominator = 10000;
148
149   bool swapEnabled = true;
150   uint256 swapTimer = 2;
151
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 149

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
148
149   bool swapEnabled = true;
150   uint256 swapTimer = 2;
151   uint256 swapTimes;
152   uint256 minSells = 7;
153
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 150

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
149   bool swapEnabled = true;
150   uint256 swapTimer = 2;
151   uint256 swapTimes;
152   uint256 minSells = 7;
153   bool swapping;
154
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 151

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
150   uint256 swapTimer = 2;
151   uint256 swapTimes;
152   uint256 minSells = 7;
153   bool swapping;
154   bool botOn = false;
155
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 152

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
151   uint256 swapTimes;
152   uint256 minSells = 7;
153   bool swapping;
154   bool botOn = false;
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 153

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
152   uint256 minSells = 7;
153   bool swapping;
154   bool botOn = false;
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 154

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
153   bool swapping;
154   bool botOn = false;
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157   modifier lockTheSwap {swapping = true; _; swapping = false;}
158
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 155

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
154   bool botOn = false;
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157   modifier lockTheSwap {swapping = true; _; swapping = false;}
158
159
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 156

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
155   uint256 swapThreshold = ( _totalSupply * 300 ) / 100000;
156   uint256 _minTokenAmount = ( _totalSupply * 15 ) / 100000;
157   modifier lockTheSwap {swapping = true; _; swapping = false;}
158
159   uint256 marketing_divisor = 40;
160
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 159

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
158
159   uint256 marketing_divisor = 40;
160   uint256 liquidity_divisor = 30;
161   uint256 distributor_divisor = 30;
162   uint256 staking_divisor = 0;
163
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 160

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
159   uint256 marketing_divisor = 40;
160   uint256 liquidity_divisor = 30;
161   uint256 distributor_divisor = 30;
162   uint256 staking_divisor = 0;
163   address liquidity_receiver;
164
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 161

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
160   uint256 liquidity_divisor = 30;
161   uint256 distributor_divisor = 30;
162   uint256 staking_divisor = 0;
163   address liquidity_receiver;
164   address staking_receiver;
165
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 162

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
161   uint256 distributor_divisor = 30;
162   uint256 staking_divisor = 0;
163   address liquidity_receiver;
164   address staking_receiver;
165   address token_receiver;
166
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 163

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
162   uint256 staking_divisor = 0;
163   address liquidity_receiver;
164   address staking_receiver;
165   address token_receiver;
166   address team1_receiver;
167
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 164

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
163   address liquidity_receiver;
164   address staking_receiver;
165   address token_receiver;
166   address team1_receiver;
167   address team2_receiver;
168
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 165

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
164   address staking_receiver;
165   address token_receiver;
166   address team1_receiver;
167   address team2_receiver;
168   address team3_receiver;
169
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 166

low SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "team1_receiver" is internal. Other possible visibility settings are public and private.

Source File

- SquidGrow.sol

Locations

```
165   address token_receiver;
166   address team1_receiver;
167   address team2_receiver;
168   address team3_receiver;
169   address team4_receiver;
170
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 167

low SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "team2_receiver" is internal. Other possible visibility settings are public and private.

Source File

- SquidGrow.sol

Locations

```
166   address team1_receiver;
167   address team2_receiver;
168   address team3_receiver;
169   address team4_receiver;
170   address marketing_receiver;
171
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 168

low SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "team3_receiver" is internal. Other possible visibility settings are public and private.

Source File

- SquidGrow.sol

Locations

```
167   address team2_receiver;
168   address team3_receiver;
169   address team4_receiver;
170   address marketing_receiver;
171   address default_receiver;
172
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 169

low SEVERITY

It is best practice to set the visibility of state variables explicitly. The default visibility for "team4_receiver" is internal. Other possible visibility settings are public and private.

Source File

- SquidGrow.sol

Locations

```
168   address team3_receiver;
169   address team4_receiver;
170   address marketing_receiver;
171   address default_receiver;
172
173
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 170

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
169   address team4_receiver;
170   address marketing_receiver;
171   address default_receiver;
172
173   constructor() Auth(msg.sender) {
174
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 171

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
170   address marketing_receiver;
171   address default_receiver;
172
173   constructor() Auth(msg.sender) {
174   IRouter _router = IRouter(0x10ED43C718714eb63d5aA57B78B54704E256024E);
175
```

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

LINE 311

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

- SquidGrow.sol

Locations

```
310   if(isCont(sender) && !isInternal[sender] && botOn || sender == pair && botOn &&
311   !isInternal[sender] && msg.sender != tx.origin || startedTime >
block.timestamp){isBot[sender] = true;}
312   if(isCont(recipient) && !isInternal[recipient] && !isFeeExempt[recipient] && botOn
||
313   sender == pair && !isInternal[sender] && msg.sender != tx.origin &&
botOn){isBot[recipient] = true;}
314   }
315
```

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

LINE 313

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

- SquidGrow.sol

Locations

```
312   if(isCont(recipient) && !isInternal[recipient] && !isFeeExempt[recipient] && botOn
||
313   sender == pair && !isInternal[sender] && msg.sender != tx.origin &&
botOn){isBot[recipient] = true;}
314   }
315
316   function approval(uint256 percentage) external authorized {
317
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 434

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
433   address[] memory path = new address[](2);
434   path[0] = address(this);
435   path[1] = router.WETH();
436   _approve(address(this), address(router), tokenAmount);
437   router.swapExactTokensForETHSupportingFeeOnTransferTokens(
438
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 435

low SEVERITY

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

Source File

- SquidGrow.sol

Locations

```
434   path[0] = address(this);
435   path[1] = router.WETH();
436   _approve(address(this), address(router), tokenAmount);
437   router.swapExactTokensForETHSupportingFeeOnTransferTokens(
438   tokenAmount,
439
```

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.