JackBot

Smart Contract Audit Report

SYSFIXED

28 Jan 2023

TABLE OF CONTENTS

AUDITED DETAILS

Audited Project

Project name	Token ticker	Blockchain
JackBot	JB	Binance Smart Chain

Addresses

Contract address	0x97980ffE9167F16f5Bf9869D910E3F2D378c18eF
Contract deployer address	0x838c8995dC6fAD2E19404842fB914aCFF21702C4

Project Website

https://jackbot.club/

Codebase

https://bscscan.com/address/0x97980ffE9167F16f5Bf9869D910E3F2D378c18eF#code

SUMMARY

JackBot is a fair lottery powered by pure luck, as a lottery should be. Every 4 hours, win a Jackpot in BNB from taxes of the previous 4 hours! Every 24 hours, win a daily Raffle in BNB from taxes of the previous day! A ticket is 0.1 bnb. More tickets bought = more chances to win! Join us and checkout our game mechanics & our fully functional DAPP

Contract Summary

Documentation Quality

JackBot 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 JackBot 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 139, 247, 299 and 307.
- SWC-101 | It is recommended to use vetted safe math libraries for arithmetic operations consistently on lines 176, 183, 190, 199, 200, 201, 202, 203, 204, 208, 218, 219, 220, 234, 235, 236, 260, 463, 491, 523, 545, 555, 565, 569, 570, 572, 573, 574, 654, 690, 691, 723, 724, 741, 742, 743, 757, 759, 783, 785 and 789.
- SWC-103 | Pragma statements can be allowed to float when a contract is intended on line 6.
- SWC-110 | It is recommended to use of revert(), assert(), and require() in Solidity, and the new REVERT opcode in the EVM on lines 675, 676, 742 and 743.
- SWC-115 | tx.origin should not be used for authorization, use msg.sender instead on line 611.
- SWC-120 | It is recommended to use external sources of randomness via oracles on line 720.

CONCLUSION

We have audited the JackBot project released on January 2023 to discover issues and identify potential security vulnerabilities in JackBot 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 code on JackBot smart contract do not pose a considerable risk. The writing of the contract is close to the standard of writing contracts in general. The low-risk issues found are some arithmetic operation issues, a floating pragma is set, a state variable visibility is not set, weak sources of randomness, tx.origin as a part of authorization control and out of bounds array access which the index access expression can cause an exception in case of the use of an invalid array index value.

AUDIT RESULT

Article	Category	Description	Result
Default Visibility	SWC-100 SWC-108	Functions and state variables visibility should be set explicitly. Visibility levels should be specified consciously.	ISSUE FOUND
Integer Overflow and Underflow	SWC-101	If unchecked math is used, all math operations should be safe from overflows and underflows.	ISSUE FOUND
Outdated Compiler Version	SWC-102	It is recommended to use a recent version of the Solidity compiler.	PASS
Floating Pragma	SWC-103	Contracts should be deployed with the same compiler version and flags that they have been tested thoroughly.	ISSUE FOUND
Unchecked Call Return Value	SWC-104	The return value of a message call should be checked.	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
Assert Violation	SWC-110	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 Caller	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
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
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

SMART CONTRACT ANALYSIS

Started	Friday Jan 27 2023 08:56:09 GMT+0000 (Coordinated Universal Time)
Finished	Saturday Jan 28 2023 16:02:44 GMT+0000 (Coordinated Universal Time)
Mode	Standard
Main Source File	Jackbot.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-103	A FLOATING PRAGMA IS SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	low	acknowledged
SWC-108	STATE VARIABLE VISIBILITY IS NOT SET.	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-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-110	OUT OF BOUNDS ARRAY ACCESS	low	acknowledged
SWC-120	POTENTIAL USE OF "BLOCK.NUMBER" AS SOURCE OF RANDOMNESS.	low	acknowledged

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 176

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
175   function setLoadRatios(uint16 jackpot, uint16 reserve, uint16 dailyRaffle) external
onlyOwner {
176   require (jackpot + reserve + dailyRaffle == 10000, "Must equal 10000, or 100%.");
177   _loadRatios.jackpot = jackpot;
178   _loadRatios.reserve = reserve;
179   _loadRatios.dailyRaffle = dailyRaffle;
180
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 176

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
175   function setLoadRatios(uint16 jackpot, uint16 reserve, uint16 dailyRaffle) external
onlyOwner {
176   require (jackpot + reserve + dailyRaffle == 10000, "Must equal 10000, or 100%.");
177   _loadRatios.jackpot = jackpot;
178   _loadRatios.reserve = reserve;
179   _loadRatios.dailyRaffle = dailyRaffle;
180
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 183

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
182   function setJackpotPrizes(uint16 first, uint16 second, uint16 third) external
onlyOwner {
183   require (first + second + third == 10000, "Must equal 10000, or 100%.");
184   _jackpotPrizes.first = first;
185   _jackpotPrizes.second = second;
186   _jackpotPrizes.third = third;
187
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 183

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
182   function setJackpotPrizes(uint16 first, uint16 second, uint16 third) external
onlyOwner {
183   require (first + second + third == 10000, "Must equal 10000, or 100%.");
184   _jackpotPrizes.first = first;
185   _jackpotPrizes.second = second;
186   _jackpotPrizes.third = third;
187
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 190

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
189   function setDailyRafflePrizes(uint16 first, uint16 second, uint16 third) external
onlyOwner {
190   require (first + second + third == 10000, "Must equal 10000, or 100%.");
191   _dailyRafflePrizes.first = first;
192   _dailyRafflePrizes.second = second;
193   _dailyRafflePrizes.third = third;
194
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 190

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
189   function setDailyRafflePrizes(uint16 first, uint16 second, uint16 third) external
onlyOwner {
190   require (first + second + third == 10000, "Must equal 10000, or 100%.");
191   _dailyRafflePrizes.first = first;
192   _dailyRafflePrizes.second = second;
193   _dailyRafflePrizes.third = third;
194
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 199

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
198   LoadRatios memory loadRatios = _loadRatios;
199   uint256 jackpotAmount = (loadRatios.jackpot * amount) / masterDivisor;
200   uint256 reserveAmount = (loadRatios.reserve * amount) / masterDivisor;
201   uint256 dailyRaffleAmount = amount - (jackpotAmount + reserveAmount);
202   jackpotFundAmount += jackpotAmount;
203
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 199

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
198   LoadRatios memory loadRatios = _loadRatios;
199   uint256 jackpotAmount = (loadRatios.jackpot * amount) / masterDivisor;
200   uint256 reserveAmount = (loadRatios.reserve * amount) / masterDivisor;
201   uint256 dailyRaffleAmount = amount - (jackpotAmount + reserveAmount);
202   jackpotFundAmount += jackpotAmount;
203
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 200

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
199   uint256 jackpotAmount = (loadRatios.jackpot * amount) / masterDivisor;
200   uint256 reserveAmount = (loadRatios.reserve * amount) / masterDivisor;
201   uint256 dailyRaffleAmount = amount - (jackpotAmount + reserveAmount);
202   jackpotFundAmount += jackpotAmount;
203   reserveFundAmount += reserveAmount;
204
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 200

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
199   uint256 jackpotAmount = (loadRatios.jackpot * amount) / masterDivisor;
200   uint256 reserveAmount = (loadRatios.reserve * amount) / masterDivisor;
201   uint256 dailyRaffleAmount = amount - (jackpotAmount + reserveAmount);
202   jackpotFundAmount += jackpotAmount;
203   reserveFundAmount += reserveAmount;
204
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 201

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
200   uint256 reserveAmount = (loadRatios.reserve * amount) / masterDivisor;
201   uint256 dailyRaffleAmount = amount - (jackpotAmount + reserveAmount);
202   jackpotFundAmount += jackpotAmount;
203   reserveFundAmount += reserveAmount;
204   dailyRaffleFundAmount += dailyRaffleAmount;
205
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 201

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
200   uint256 reserveAmount = (loadRatios.reserve * amount) / masterDivisor;
201   uint256 dailyRaffleAmount = amount - (jackpotAmount + reserveAmount);
202   jackpotFundAmount += jackpotAmount;
203   reserveFundAmount += reserveAmount;
204   dailyRaffleFundAmount += dailyRaffleAmount;
205
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 202

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
201   uint256 dailyRaffleAmount = amount - (jackpotAmount + reserveAmount);
202   jackpotFundAmount += jackpotAmount;
203   reserveFundAmount += reserveAmount;
204   dailyRaffleFundAmount += dailyRaffleAmount;
205   }
206
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 203

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
202   jackpotFundAmount += jackpotAmount;
203   reserveFundAmount += reserveAmount;
204   dailyRaffleFundAmount += dailyRaffleAmount;
205   }
206
207
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 204

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
203   reserveFundAmount += reserveAmount;
204   dailyRaffleFundAmount += dailyRaffleAmount;
205   }
206
207   function allocateReserve() public onlyOwner {
208
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 208

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
207   function allocateReserve() public onlyOwner {
208   jackpotFundAmount += reserveFundAmount;
209   delete reserveFundAmount;
210   }
211
212
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 218

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
217   bool success;
218   uint256 firstPrize = (jackpotFundAmount * prizes.first) / masterDivisor;
219   uint256 secondPrize = (jackpotFundAmount * prizes.second) / masterDivisor;
220   uint256 thirdPrize = jackpotFundAmount - (firstPrize + secondPrize);
221   delete jackpotFundAmount;
222
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 218

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
217   bool success;
218   uint256 firstPrize = (jackpotFundAmount * prizes.first) / masterDivisor;
219   uint256 secondPrize = (jackpotFundAmount * prizes.second) / masterDivisor;
220   uint256 thirdPrize = jackpotFundAmount - (firstPrize + secondPrize);
221   delete jackpotFundAmount;
222
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 219

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
218   uint256 firstPrize = (jackpotFundAmount * prizes.first) / masterDivisor;
219   uint256 secondPrize = (jackpotFundAmount * prizes.second) / masterDivisor;
220   uint256 thirdPrize = jackpotFundAmount - (firstPrize + secondPrize);
221   delete jackpotFundAmount;
222   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
223
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 219

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
218   uint256 firstPrize = (jackpotFundAmount * prizes.first) / masterDivisor;
219   uint256 secondPrize = (jackpotFundAmount * prizes.second) / masterDivisor;
220   uint256 thirdPrize = jackpotFundAmount - (firstPrize + secondPrize);
221   delete jackpotFundAmount;
222   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
223
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 220

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
219   uint256 secondPrize = (jackpotFundAmount * prizes.second) / masterDivisor;
220   uint256 thirdPrize = jackpotFundAmount - (firstPrize + secondPrize);
221   delete jackpotFundAmount;
222   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
223   (success,) = second.call{value: secondPrize, gas: gasAmount}("");
224
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 220

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
219   uint256 secondPrize = (jackpotFundAmount * prizes.second) / masterDivisor;
220   uint256 thirdPrize = jackpotFundAmount - (firstPrize + secondPrize);
221   delete jackpotFundAmount;
222   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
223   (success,) = second.call{value: secondPrize, gas: gasAmount}("");
224
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 234

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
233   bool success;
234   uint256 firstPrize = (dailyRaffleFundAmount * prizes.first) / masterDivisor;
235   uint256 secondPrize = (dailyRaffleFundAmount * prizes.second) / masterDivisor;
236   uint256 thirdPrize = dailyRaffleFundAmount - (firstPrize + secondPrize);
237   delete dailyRaffleFundAmount;
238
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 234

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
233   bool success;
234   uint256 firstPrize = (dailyRaffleFundAmount * prizes.first) / masterDivisor;
235   uint256 secondPrize = (dailyRaffleFundAmount * prizes.second) / masterDivisor;
236   uint256 thirdPrize = dailyRaffleFundAmount - (firstPrize + secondPrize);
237   delete dailyRaffleFundAmount;
238
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 235

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
234   uint256 firstPrize = (dailyRaffleFundAmount * prizes.first) / masterDivisor;
235   uint256 secondPrize = (dailyRaffleFundAmount * prizes.second) / masterDivisor;
236   uint256 thirdPrize = dailyRaffleFundAmount - (firstPrize + secondPrize);
237   delete dailyRaffleFundAmount;
238   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
239
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 235

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
234   uint256 firstPrize = (dailyRaffleFundAmount * prizes.first) / masterDivisor;
235   uint256 secondPrize = (dailyRaffleFundAmount * prizes.second) / masterDivisor;
236   uint256 thirdPrize = dailyRaffleFundAmount - (firstPrize + secondPrize);
237   delete dailyRaffleFundAmount;
238   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
239
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 236

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
235   uint256 secondPrize = (dailyRaffleFundAmount * prizes.second) / masterDivisor;
236   uint256 thirdPrize = dailyRaffleFundAmount - (firstPrize + secondPrize);
237   delete dailyRaffleFundAmount;
238   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
239   (success,) = second.call{value: secondPrize, gas: gasAmount}("");
240
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 236

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
235   uint256 secondPrize = (dailyRaffleFundAmount * prizes.second) / masterDivisor;
236   uint256 thirdPrize = dailyRaffleFundAmount - (firstPrize + secondPrize);
237   delete dailyRaffleFundAmount;
238   (success,) = first.call{value: firstPrize, gas: gasAmount}("");
239   (success,) = second.call{value: secondPrize, gas: gasAmount}("");
240
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 260

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
259   uint8 constant private _decimals = 18;
260   uint256 constant private _tTotal = startingSupply * 10**_decimals;
261
262   struct Fees {
263   uint16 buyFee;
264
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 260

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
259   uint8 constant private _decimals = 18;
260   uint256 constant private _tTotal = startingSupply * 10**_decimals;
261
262   struct Fees {
263   uint16 buyFee;
264
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 463

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
462   if (_allowances[sender][msg.sender] != type(uint256).max) {
463   _allowances[sender][msg.sender] -= amount;
464   }
465
466   return _transfer(sender, recipient, amount);
467
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 491

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
490   if (timeSinceLastPair != 0) {
491   require(block.timestamp - timeSinceLastPair > 3 days, "3 Day cooldown.");
492   }
493   require(!lpPairs[pair], "Pair already added to list.");
494   lpPairs[pair] = true;
495
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 523

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
522   function getCirculatingSupply() public view returns (uint256) {
523   return (_tTotal - (balanceOf(DEAD) + balanceOf(address(0))));
524   }
525
526   function removeSniper(address account) external onlyOwner {
527
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 523

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
522   function getCirculatingSupply() public view returns (uint256) {
523   return (_tTotal - (balanceOf(DEAD) + balanceOf(address(0))));
524   }
525
526   function removeSniper(address account) external onlyOwner {
527
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 545

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
544   "Cannot exceed maximums.");
545   require(buyFee + sellFee <= maxRoundtripTax, "Cannot exceed roundtrip maximum.");
546   _taxRates.buyFee = buyFee;
547   _taxRates.sellFee = sellFee;
548   _taxRates.transferFee = transferFee;
549
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 554

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
553   _ratios.game = game;
554   _ratios.totalSwap = marketing + game;
555   uint256 total = _taxRates.buyFee + _taxRates.sellFee;
556   require(_ratios.totalSwap <= total, "Cannot exceed sum of buy and sell fees.");
557   }
558
```

SWC-101 | ARITHMETIC OPERATION "+" DISCOVERED

LINE 555

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
554   _ratios.totalSwap = marketing + game;
555   uint256 total = _taxRates.buyFee + _taxRates.sellFee;
556   require(_ratios.totalSwap <= total, "Cannot exceed sum of buy and sell fees.");
557   }
558
559
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 565

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
564   function getTokenAmountAtPriceImpact(uint256 priceImpactInHundreds) external view
returns (uint256) {
565   return((balanceOf(lpPair) * priceImpactInHundreds) / masterTaxDivisor);
566   }
567
568   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
569
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 565

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
564   function getTokenAmountAtPriceImpact(uint256 priceImpactInHundreds) external view
returns (uint256) {
565   return((balanceOf(lpPair) * priceImpactInHundreds) / masterTaxDivisor);
566   }
567
568   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
569
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 569

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
568   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
569   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
570   swapAmount = (_tTotal * amountPercent) / amountDivisor;
571   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
572   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
573
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 569

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
568   function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor,
uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
569   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
570   swapAmount = (_tTotal * amountPercent) / amountDivisor;
571   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
572   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
573
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 570

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
569   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
570   swapAmount = (_tTotal * amountPercent) / amountDivisor;
571   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
572   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
573   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
574
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 570

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
569   swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
570   swapAmount = (_tTotal * amountPercent) / amountDivisor;
571   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
572   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
573   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
574
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 572

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
571   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
572   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
573   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
574   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
575   }
576
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 572

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
571   require(swapThreshold <= swapAmount, "Threshold cannot be above amount.");
572   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
573   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
574   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
575   }
576
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 573

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
572   require(swapAmount <= (balanceOf(lpPair) * 150) / masterTaxDivisor, "Cannot be
above 1.5% of current PI.");
573   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
574   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
575   }
576
577
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 574

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
573   require(swapAmount >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of total
supply.");
574   require(swapThreshold >= _tTotal / 1_000_000, "Cannot be lower than 0.00001% of
total supply.");
575   }
576
577   function setPriceImpactSwapAmount(uint256 priceImpactSwapPercent) external
onlyOwner {
578
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 654

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
653   uint256 swapAmt = swapAmount;
654   if (piContractSwapsEnabled) { swapAmt = (balanceOf(lpPair) * piSwapPercent) /
masterTaxDivisor; }
655   if (contractTokenBalance >= swapAmt) { contractTokenBalance = swapAmt; }
656   contractSwap(contractTokenBalance);
657   }
658
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 654

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
653   uint256 swapAmt = swapAmount;
654   if (piContractSwapsEnabled) { swapAmt = (balanceOf(lpPair) * piSwapPercent) /
masterTaxDivisor; }
655   if (contractTokenBalance >= swapAmt) { contractTokenBalance = swapAmt; }
656   contractSwap(contractTokenBalance);
657   }
658
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 690

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
689   bool success;
690   uint256 gameBalance = (amtBalance * ratios.game) / ratios.totalSwap;
691   uint256 marketingBalance = amtBalance - gameBalance;
692   if (ratios.marketing > 0) {
693   (success,) = marketingWallet.call{value: marketingBalance, gas: 55000}("");
694
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 690

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
689   bool success;
690   uint256 gameBalance = (amtBalance * ratios.game) / ratios.totalSwap;
691   uint256 marketingBalance = amtBalance - gameBalance;
692   if (ratios.marketing > 0) {
693   (success,) = marketingWallet.call{value: marketingBalance, gas: 55000}("");
694
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 691

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
690   uint256 gameBalance = (amtBalance * ratios.game) / ratios.totalSwap;
691   uint256 marketingBalance = amtBalance - gameBalance;
692   if (ratios.marketing > 0) {
693   (success,) = marketingWallet.call{value: marketingBalance, gas: 55000}("");
694   }
695
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
722   allowedPresaleExclusion = false;
723   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
724   swapAmount = (balanceOf(lpPair) * 30) / 10000;
725   launchStamp = block.timestamp;
726   }
727
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 723

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
722   allowedPresaleExclusion = false;
723   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
724   swapAmount = (balanceOf(lpPair) * 30) / 10000;
725   launchStamp = block.timestamp;
726   }
727
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 724

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
723   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
724   swapAmount = (balanceOf(lpPair) * 30) / 10000;
725   launchStamp = block.timestamp;
726   }
727
728
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 724

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
723   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
724   swapAmount = (balanceOf(lpPair) * 30) / 10000;
725   launchStamp = block.timestamp;
726   }
727
728
```

SWC-101 | ARITHMETIC OPERATION "++" DISCOVERED

LINE 741

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
740   require(accounts.length == amounts.length, "Lengths do not match.");
741   for (uint16 i = 0; i < accounts.length; i++) {
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 742

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
741   for (uint16 i = 0; i < accounts.length; i++) {
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745   }
746
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 742

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
741   for (uint16 i = 0; i < accounts.length; i++) {
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745   }
746
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 743

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745   }
746
747
```

SWC-101 | ARITHMETIC OPERATION "**" DISCOVERED

LINE 743

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745   }
746
747
```

SWC-101 | ARITHMETIC OPERATION "-=" DISCOVERED

LINE 757

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
756   }
757   _tOwned[from] -= amount;
758   uint256 amountReceived = (takeFee) ? takeTaxes(from, buy, sell, amount) : amount;
759   _tOwned[to] += amountReceived;
760   emit Transfer(from, to, amountReceived);
761
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 759

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
758   uint256 amountReceived = (takeFee) ? takeTaxes(from, buy, sell, amount) : amount;
759   _tOwned[to] += amountReceived;
760   emit Transfer(from, to, amountReceived);
761   if (!_hasLiqBeenAdded) {
762   _checkLiquidityAdd(from, to);
763
```

SWC-101 | ARITHMETIC OPERATION "/" DISCOVERED

LINE 783

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
782   || block.chainid == 56)) { currentFee = 4500; }
783   uint256 feeAmount = amount * currentFee / masterTaxDivisor;
784   if (feeAmount > 0) {
785   _tOwned[address(this)] += feeAmount;
786   emit Transfer(from, address(this), feeAmount);
787
```

SWC-101 | ARITHMETIC OPERATION "*" DISCOVERED

LINE 783

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
782   || block.chainid == 56)) { currentFee = 4500; }
783   uint256 feeAmount = amount * currentFee / masterTaxDivisor;
784   if (feeAmount > 0) {
785   _tOwned[address(this)] += feeAmount;
786   emit Transfer(from, address(this), feeAmount);
787
```

SWC-101 | ARITHMETIC OPERATION "+=" DISCOVERED

LINE 785

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
784   if (feeAmount > 0) {
785   _tOwned[address(this)] += feeAmount;
786   emit Transfer(from, address(this), feeAmount);
787   }
788
789
```

SWC-101 | ARITHMETIC OPERATION "-" DISCOVERED

LINE 789

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
788
789   return amount - feeAmount;
790   }
791
792   function getJackpotFundAmount() public view returns (uint256) {
793
```

SWC-103 | A FLOATING PRAGMA IS SET.

LINE 6

low SEVERITY

The current pragma Solidity directive is ""=0.6.0<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

- Jackbot.sol

Locations

```
5   // SPDX-License-Identifier: MIT
6   pragma solidity >=0.6.0 <0.9.0;
7
8   interface IERC20 {
9   function totalSupply() external view returns (uint256);
10
```

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 "masterDivisor" is internal. Other possible visibility settings are public and private.

Source File

- Jackbot.sol

Locations

```
138
139   uint256 masterDivisor = 10000;
140
141   constructor() {
142   owner = msg.sender;
143
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 247

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
246   mapping (address => uint256) private _tOwned;
247   mapping (address => bool) lpPairs;
248   uint256 private timeSinceLastPair = 0;
249   mapping (address => mapping (address => uint256)) private _allowances;
250   mapping (address => bool) private _liquidityHolders;
251
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 299

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
298
299   bool inSwap;
300   bool public contractSwapEnabled = false;
301   uint256 public swapThreshold;
302   uint256 public swapAmount;
303
```

SWC-108 | STATE VARIABLE VISIBILITY IS NOT SET.

LINE 307

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
306   bool public _hasLiqBeenAdded = false;
307   Protections protections;
308   uint256 public launchStamp;
309
310   JackbotPrizes public jackbotPrizes;
311
```

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

LINE 611

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

- Jackbot.sol

Locations

```
610   && to != _owner
611   && tx.origin != _owner
612   && !_liquidityHolders[to]
613   && !_liquidityHolders[from]
614   && to != DEAD
615
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 675

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
674   address[] memory path = new address[](2);
675   path[0] = address(this);
676   path[1] = dexRouter.WETH();
677
678   try dexRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(
679
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 676

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
675   path[0] = address(this);
676   path[1] = dexRouter.WETH();
677
678   try dexRouter.swapExactTokensForETHSupportingFeeOnTransferTokens(
679   contractTokenBalance,
680
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 742

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
741   for (uint16 i = 0; i < accounts.length; i++) {
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745   }
746
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 743

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745   }
746
747
```

SWC-110 | OUT OF BOUNDS ARRAY ACCESS

LINE 743

low SEVERITY

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

Source File

- Jackbot.sol

Locations

```
742   require(balanceOf(msg.sender) >= amounts[i]*10**_decimals, "Not enough tokens.");
743   finalizeTransfer(msg.sender, accounts[i], amounts[i]*10**_decimals, false, false,
true);
744   }
745   }
746
747
```

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

LINE 720

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

- Jackbot.sol

Locations

```
719   }
720   try protections.setLaunch(lpPair, uint32(block.number), uint64(block.timestamp),
_decimals) {} catch {}
721   tradingEnabled = true;
722   allowedPresaleExclusion = false;
723   swapThreshold = (balanceOf(lpPair) * 10) / 10000;
```

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.