RUGDOG

SMART CONTRACT SECURITY AUDIT

Final report

Plan: Simple

GOAT Coin

January 2024

rugdog.net

the@rugdog.net

✧ CONTENTS

✧ INTRODUCTION

The report has been prepared for GOAT Coin.

A dividend token made with launchpad. The functionality includes automated liquidity and rewards, fees and exclusions, antibot features.

Name	GOAT Coin
Audit date	2024-01-13 - 2024-01-13
Language	Solidity
Network	Binance Smart Chain

✧ CONTRACTS CHECKED

Name	Address
AntiBotBABYTOKEN	0x2BD7b29035Ec9E110D65B063B5Ce84F328d8685d

✧ AUDIT PROCESS

The code was audited by the team according to the following order:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual confirmation of all the issues found by the tools

Manual audit

- Thorough manual analysis of smart contracts for security vulnerabilities
- Smart contracts' logic check

ATTACKS CHECKED

Title	Check result
Unencrypted Private Data On-Chain	✓ passed
Code With No Effects	✓ passed
Message call with hardcoded gas amount	✓ passed
Typographical Error	✓ passed
DoS With Block Gas Limit	✓ passed
Presence of unused variables	✓ passed
Incorrect Inheritance Order	✓ passed
Requirement Violation	✓ passed
Weak Sources of Randomness from Chain Attributes	✓ passed
Shadowing State Variables	✓ passed
Incorrect Constructor Name	✓ passed
Block values as a proxy for time	✓ passed
Authorization through tx.origin	✓ passed

DoS with Failed Call	✓ passed
Delegatecall to Untrusted Callee	✓ passed
Use of Deprecated Solidity Functions	✓ passed
Assert Violation	✓ passed
State Variable Default Visibility	✓ passed
Reentrancy	✓ passed
Unprotected SELFDESTRUCT Instruction	✓ passed
Unprotected Ether Withdrawal	✓ passed
Unchecked Call Return Value	✓ passed
Floating Pragma	✓ passed
Outdated Compiler Version	✓ passed
Integer Overflow and Underflow	✓ passed
Function Default Visibility	✓ passed

✧ OVERVIEW OF RELEVANCE LEVELS

High relevance Issues of high relevance may lead to losses of users' funds as well as changes of ownership of a contract or possible issues with the logic of the contract.
High-relevance issues require immediate attention and a response from the team.

Medium relevance	While issues of medium relevance don't pose as high a risk as the high-relevance ones do, they can be just as easily exploited by the team or a malicious user, causing a contract failure and damaging the project's reputation in the process. Usually, these issues can be fixed if the contract is redeployed. Medium-relevance issues require a response from the team.
Low relevance	Issues of low relevance don't pose high risks since they can't cause damage to the functionality of the contract. However, it's still recommended to consider fixing them.

ISSUES

High relevance issues

No high relevance issues found

Medium relevance issues

No medium relevance issues found

Low relevance issues

No low relevance issues found

✧ CONCLUSION

GOAT Coin AntiBotBABYTOKEN contract was audited. No relevance issues were found.

✧ 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 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 without RugDog prior written consent.

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 RugDog 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 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.
The rights to publish the results of this audit are exclusively retained by RugDog.

✧ AUTOMATED ANALYSIS

```
AntiBotBABYTOKEN.addLiquidity(uint256,uint256) (goattoken.sol#3472-3485) sends eth
to arbitrary user
▐Dangerous calls:
▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#functions-
that-send-ether-to-arbitrary-destinations

Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐▐- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▐- swapAndSendDividends(sellTokens) (goattoken.sol#3354)
▐▐- (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
▐▐- success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),address(dividendTrac
ker),dividends) (goattoken.sol#3490-3494)
```

```
▐▐- dividendTracker.distributeCAKEDividends(dividends) (goattoken.sol#3496)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐State variables written after the call(s):
▐- super._transfer(from,address(this),fees) (goattoken.sol#3371)
▐▐- _balances[sender] = senderBalance - amount (goattoken.sol#379)
▐▐- _balances[recipient] += amount (goattoken.sol#381)
▐- super._transfer(from,to,amount) (goattoken.sol#3374)
▐▐- _balances[sender] = senderBalance - amount (goattoken.sol#379)
▐▐- _balances[recipient] += amount (goattoken.sol#381)
▐- swapping = false (goattoken.sol#3357)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities

Reentrancy in DividendPayingToken._withdrawDividendOfUser(address)
(goattoken.sol#2506-2533):
▐External calls:
▐- success =
SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),user,_withdrawableDividend)
(goattoken.sol#2517-2521)
▐State variables written after the call(s):
▐- withdrawnDividends[user] = withdrawnDividends[user].sub(_withdrawableDividend)
(goattoken.sol#2524-2526)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-1

AntiBotBABYTOKEN._transfer(address,address,uint256).iterations (goattoken.sol#3385)
is a local variable never initialized
AntiBotBABYTOKEN._transfer(address,address,uint256).claims (goattoken.sol#3386) is a
local variable never initialized
```

```
AntiBotBABYTOKEN._transfer(address,address,uint256).lastProcessedIndex
(goattoken.sol#3387) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-
Documentation#uninitialized-local-variables

AntiBotBABYTOKEN.claim() (goattoken.sol#3295-3297) ignores return value by
dividendTracker.processAccount(address(msg.sender),false) (goattoken.sol#3296)
AntiBotBABYTOKEN._transfer(address,address,uint256) (goattoken.sol#3307-3399)
ignores return value by dividendTracker.process(gas) (goattoken.sol#3384-3397)
AntiBotBABYTOKEN.addLiquidity(uint256,uint256) (goattoken.sol#3472-3485) ignores
return value by uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-
return

DividendPayingToken.__DividendPayingToken_init(address,string,string)._name
(goattoken.sol#2477) shadows:
▌- ERC20Upgradeable._name (goattoken.sol#1742) (state variable)
DividendPayingToken.__DividendPayingToken_init(address,string,string)._symbol
(goattoken.sol#2478) shadows:
▌- ERC20Upgradeable._symbol (goattoken.sol#1743) (state variable)
DividendPayingToken.dividendOf(address)._owner (goattoken.sol#2538) shadows:
▌- OwnableUpgradeable._owner (goattoken.sol#2086) (state variable)
DividendPayingToken.withdrawableDividendOf(address)._owner (goattoken.sol#2545)
shadows:
▌- OwnableUpgradeable._owner (goattoken.sol#2086) (state variable)
DividendPayingToken.withdrawnDividendOf(address)._owner (goattoken.sol#2557)
shadows:
▌- OwnableUpgradeable._owner (goattoken.sol#2086) (state variable)
DividendPayingToken.accumulativeDividendOf(address)._owner (goattoken.sol#2571)
shadows:
▌- OwnableUpgradeable._owner (goattoken.sol#2086) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-
variable-shadowing

AntiBotBABYTOKEN.setSwapTokensAtAmount(uint256) (goattoken.sol#3103-3109) should
```

```
emit an event for:
	- swapTokensAtAmount = amount (goattoken.sol#3108)
AntiBotBABYTOKEN.setTokenRewardsFee(uint256) (goattoken.sol#3141-3145) should emit
an event for:
	- totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee)
(goattoken.sol#3143)
AntiBotBABYTOKEN.setLiquiditFee(uint256) (goattoken.sol#3147-3151) should emit an
event for:
	- liquidityFee = value (goattoken.sol#3148)
	- totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee)
(goattoken.sol#3149)
AntiBotBABYTOKEN.setMarketingFee(uint256) (goattoken.sol#3153-3157) should emit an
event for:
	- marketingFee = value (goattoken.sol#3154)
	- totalFees = tokenRewardsFee.add(liquidityFee).add(marketingFee)
(goattoken.sol#3155)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-
events-arithmetic

AntiBotBABYTOKEN.constructor(string,string,uint256,address[5],uint256[3],uint256,addr
ess,uint256)._uniswapV2Pair (goattoken.sol#3065-3066) lacks a zero-check on :
	- uniswapV2Pair = _uniswapV2Pair (goattoken.sol#3068)
AntiBotBABYTOKEN.constructor(string,string,uint256,address[5],uint256[3],uint256,addr
ess,uint256).serviceFeeReceiver_ (goattoken.sol#3027) lacks a zero-check on :
	- address(serviceFeeReceiver_).transfer(serviceFee_) (goattoken.sol#3094)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#missing-
zero-address-validation

SafeERC20NoRevert.safeTransfer(IERC20,address,uint256) (goattoken.sol#1226-1238) has
external calls inside a loop: (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation/#calls-
inside-a-loop

Variable 'AntiBotBABYTOKEN._transfer(address,address,uint256).claims
(goattoken.sol#3386)' in AntiBotBABYTOKEN._transfer(address,address,uint256)
```

```
(goattoken.sol#3307-3399) potentially used before declaration:
ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin)
(goattoken.sol#3389-3396)
Variable 'AntiBotBABYTOKEN._transfer(address,address,uint256).iterations
(goattoken.sol#3385)' in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399) potentially used before declaration:
ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin)
(goattoken.sol#3389-3396)
Variable 'AntiBotBABYTOKEN._transfer(address,address,uint256).lastProcessedIndex
(goattoken.sol#3387)' in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399) potentially used before declaration:
ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin)
(goattoken.sol#3389-3396)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-
declaration-usage-of-local-variables

Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐State variables written after the call(s):
▐- super._transfer(from,to,0) (goattoken.sol#3320)
▐▐- _balances[sender] = senderBalance - amount (goattoken.sol#379)
▐▐- _balances[recipient] += amount (goattoken.sol#381)
▐- swapping = true (goattoken.sol#3336)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
```

```
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐State variables written after the call(s):
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- _allowances[owner][spender] = amount (goattoken.sol#458)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐▐- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐State variables written after the call(s):
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- _allowances[owner][spender] = amount (goattoken.sol#458)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
```

```
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐▐- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▐- swapAndSendDividends(sellTokens) (goattoken.sol#3354)
▐▐- (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
▐▐- success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),address(dividendTrac
ker),dividends) (goattoken.sol#3490-3494)
▐▐- dividendTracker.distributeCAKEDividends(dividends) (goattoken.sol#3496)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐State variables written after the call(s):
▐- swapAndSendDividends(sellTokens) (goattoken.sol#3354)
▐▐- _allowances[owner][spender] = amount (goattoken.sol#458)
Reentrancy in BABYTOKENDividendTracker.processAccount(address,bool)
(goattoken.sol#2893-2907):
▐External calls:
```

```
▌- amount = _withdrawDividendOfUser(account) (goattoken.sol#2898)
▌▌- (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
▌▌- success =
SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),user,_withdrawableDividend)
(goattoken.sol#2517-2521)
▌State variables written after the call(s):
▌- lastClaimTimes[account] = block.timestamp (goattoken.sol#2901)
Reentrancy in AntiBotBABYTOKEN.swapAndLiquify(uint256) (goattoken.sol#3413-3434):
▌External calls:
▌- swapTokensForEth(half) (goattoken.sol#3425)
▌▌- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▌- addLiquidity(otherHalf,newBalance) (goattoken.sol#3431)
▌▌- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▌External calls sending eth:
▌- addLiquidity(otherHalf,newBalance) (goattoken.sol#3431)
▌▌- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▌State variables written after the call(s):
▌- addLiquidity(otherHalf,newBalance) (goattoken.sol#3431)
▌▌- _allowances[owner][spender] = amount (goattoken.sol#458)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-2

Reentrancy in AntiBotBABYTOKEN._setAutomatedMarketMakerPair(address,bool)
(goattoken.sol#3159-3171):
▌External calls:
▌- dividendTracker.excludeFromDividends(pair) (goattoken.sol#3167)
▌Event emitted after the call(s):
▌- SetAutomatedMarketMakerPair(pair,value) (goattoken.sol#3170)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
```

```
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐Event emitted after the call(s):
▐- Transfer(sender,recipient,amount) (goattoken.sol#383)
▐▐- super._transfer(from,to,0) (goattoken.sol#3320)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐Event emitted after the call(s):
▐- Approval(owner,spender,amount) (goattoken.sol#459)
▐▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
```

```
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐▐- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐Event emitted after the call(s):
▐- Approval(owner,spender,amount) (goattoken.sol#459)
▐▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐- SwapAndLiquify(half,newBalance,otherHalf) (goattoken.sol#3433)
▐▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐▐- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▐- swapAndSendDividends(sellTokens) (goattoken.sol#3354)
▐▐- (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
```

```
▐▐- success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),address(dividendTrac
ker),dividends) (goattoken.sol#3490-3494)
▐▐- dividendTracker.distributeCAKEDividends(dividends) (goattoken.sol#3496)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐Event emitted after the call(s):
▐- Approval(owner,spender,amount) (goattoken.sol#459)
▐▐- swapAndSendDividends(sellTokens) (goattoken.sol#3354)
▐- SendDividends(tokens,dividends) (goattoken.sol#3497)
▐▐- swapAndSendDividends(sellTokens) (goattoken.sol#3354)
▐- Transfer(sender,recipient,amount) (goattoken.sol#383)
▐▐- super._transfer(from,address(this),fees) (goattoken.sol#3371)
▐- Transfer(sender,recipient,amount) (goattoken.sol#383)
▐▐- super._transfer(from,to,amount) (goattoken.sol#3374)
Reentrancy in AntiBotBABYTOKEN._transfer(address,address,uint256)
(goattoken.sol#3307-3399):
▐External calls:
▐- pinkAntiBot.onPreTransferCheck(from,to,amount) (goattoken.sol#3316)
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- returndata = address(token).functionCall(data,SafeERC20: low-level call failed)
(goattoken.sol#815)
▐▐- IERC20(rewardToken).safeTransfer(_marketingWalletAddress,newBalance)
(goattoken.sol#3410)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
```

```
▐▐- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▐- swapAndSendDividends(sellTokens) (goattoken.sol#3354)
▐▐- (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
▐▐- success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),address(dividendTrac
ker),dividends) (goattoken.sol#3490-3494)
▐▐- dividendTracker.distributeCAKEDividends(dividends) (goattoken.sol#3496)
▐▐- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▐- dividendTracker.setBalance(address(from),balanceOf(from))
(goattoken.sol#3376-3378)
▐- dividendTracker.setBalance(address(to),balanceOf(to)) (goattoken.sol#3379)
▐- dividendTracker.process(gas) (goattoken.sol#3384-3397)
▐External calls sending eth:
▐- swapAndSendToFee(marketingTokens) (goattoken.sol#3342)
▐▐- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
▐- swapAndLiquify(swapTokens) (goattoken.sol#3349)
▐▐- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▐Event emitted after the call(s):
▐- ProcessedDividendTracker(iterations,claims,lastProcessedIndex,true,gas,tx.origin)
(goattoken.sol#3389-3396)
Reentrancy in BABYTOKENDividendTracker.processAccount(address,bool)
(goattoken.sol#2893-2907):
▐External calls:
▐- amount = _withdrawDividendOfUser(account) (goattoken.sol#2898)
▐▐- (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
▐▐- success =
SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),user,_withdrawableDividend)
(goattoken.sol#2517-2521)
▐Event emitted after the call(s):
▐- Claim(account,amount,automatic) (goattoken.sol#2902)
```

```
Reentrancy in AntiBotBABYTOKEN.processDividendTracker(uint256)
(goattoken.sol#3279-3293):
▌External calls:
▌- (iterations,claims,lastProcessedIndex) = dividendTracker.process(gas)
(goattoken.sol#3280-3284)
▌Event emitted after the call(s):
▌-
ProcessedDividendTracker(iterations,claims,lastProcessedIndex,false,gas,tx.origin)
(goattoken.sol#3285-3292)
Reentrancy in AntiBotBABYTOKEN.swapAndLiquify(uint256) (goattoken.sol#3413-3434):
▌External calls:
▌- swapTokensForEth(half) (goattoken.sol#3425)
▌▌- uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,
path,address(this),block.timestamp) (goattoken.sol#3445-3451)
▌- addLiquidity(otherHalf,newBalance) (goattoken.sol#3431)
▌▌- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▌External calls sending eth:
▌- addLiquidity(otherHalf,newBalance) (goattoken.sol#3431)
▌▌- uniswapV2Router.addLiquidityETH{value: ethAmount}
(address(this),tokenAmount,0,0,address(0xdead),block.timestamp)
(goattoken.sol#3477-3484)
▌Event emitted after the call(s):
▌- Approval(owner,spender,amount) (goattoken.sol#459)
▌▌- addLiquidity(otherHalf,newBalance) (goattoken.sol#3431)
▌- SwapAndLiquify(half,newBalance,otherHalf) (goattoken.sol#3433)
Reentrancy in AntiBotBABYTOKEN.swapAndSendDividends(uint256)
(goattoken.sol#3487-3499):
▌External calls:
▌- swapTokensForCake(tokens) (goattoken.sol#3488)
▌▌- uniswapV2Router.swapExactTokensForTokensSupportingFeeOnTransferTokens(tokenAmount
,0,path,address(this),block.timestamp) (goattoken.sol#3463-3469)
▌- success = SafeERC20NoRevert.safeTransfer(IERC20(rewardToken),address(dividendTrack
er),dividends) (goattoken.sol#3490-3494)
▌- dividendTracker.distributeCAKEDividends(dividends) (goattoken.sol#3496)
▌Event emitted after the call(s):
```

```
▌- SendDividends(tokens,dividends) (goattoken.sol#3497)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-
vulnerabilities-3

BABYTOKENDividendTracker.getAccount(address) (goattoken.sol#2742-2789) uses
timestamp for comparisons
▌Dangerous comparisons:
▌- nextClaimTime > block.timestamp (goattoken.sol#2786-2788)
BABYTOKENDividendTracker.canAutoClaim(uint256) (goattoken.sol#2814-2820) uses
timestamp for comparisons
▌Dangerous comparisons:
▌- lastClaimTime > block.timestamp (goattoken.sol#2815)
▌- block.timestamp.sub(lastClaimTime) >= claimWait (goattoken.sol#2819)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-
timestamp

Address.isContract(address) (goattoken.sol#530-540) uses assembly
▌- INLINE ASM (goattoken.sol#536-538)
Address.verifyCallResult(bool,bytes,string) (goattoken.sol#699-719) uses assembly
▌- INLINE ASM (goattoken.sol#711-714)
Clones.clone(address) (goattoken.sol#1151-1160) uses assembly
▌- INLINE ASM (goattoken.sol#1152-1158)
Clones.cloneDeterministic(address,bytes32) (goattoken.sol#1169-1178) uses assembly
▌- INLINE ASM (goattoken.sol#1170-1176)
Clones.predictDeterministicAddress(address,bytes32,address)
(goattoken.sol#1183-1198) uses assembly
▌- INLINE ASM (goattoken.sol#1188-1197)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-
usage

Address.functionCall(address,bytes) (goattoken.sol#583-585) is never used and should
be removed
Address.functionCallWithValue(address,bytes,uint256) (goattoken.sol#612-618) is
never used and should be removed
Address.functionDelegateCall(address,bytes) (goattoken.sol#672-674) is never used
and should be removed
Address.functionDelegateCall(address,bytes,string) (goattoken.sol#682-691) is never
```

```
used and should be removed
Address.functionStaticCall(address,bytes) (goattoken.sol#645-647) is never used and
should be removed
Address.functionStaticCall(address,bytes,string) (goattoken.sol#655-664) is never
used and should be removed
Address.sendValue(address,uint256) (goattoken.sol#558-563) is never used and should
be removed
Clones.cloneDeterministic(address,bytes32) (goattoken.sol#1169-1178) is never used
and should be removed
Clones.predictDeterministicAddress(address,bytes32) (goattoken.sol#1203-1209) is
never used and should be removed
Clones.predictDeterministicAddress(address,bytes32,address)
(goattoken.sol#1183-1198) is never used and should be removed
Context._msgData() (goattoken.sol#140-142) is never used and should be removed
ContextUpgradeable.__Context_init() (goattoken.sol#1683-1685) is never used and
should be removed
ContextUpgradeable._msgData() (goattoken.sol#1693-1695) is never used and should be
removed
DividendPayingToken._transfer(address,address,uint256) (goattoken.sol#2590-2605) is
never used and should be removed
ERC20._burn(address,uint256) (goattoken.sol#420-435) is never used and should be
removed
ERC20Upgradeable._transfer(address,address,uint256) (goattoken.sol#1926-1946) is
never used and should be removed
SafeERC20.safeApprove(IERC20,address,uint256) (goattoken.sol#767-780) is never used
and should be removed
SafeERC20.safeDecreaseAllowance(IERC20,address,uint256) (goattoken.sol#791-802) is
never used and should be removed
SafeERC20.safeIncreaseAllowance(IERC20,address,uint256) (goattoken.sol#782-789) is
never used and should be removed
SafeERC20.safeTransferFrom(IERC20,address,address,uint256) (goattoken.sol#751-758)
is never used and should be removed
SafeMath.div(uint256,uint256,string) (goattoken.sol#1088-1097) is never used and
should be removed
SafeMath.mod(uint256,uint256) (goattoken.sol#1048-1050) is never used and should be
removed
SafeMath.mod(uint256,uint256,string) (goattoken.sol#1114-1123) is never used and
```

```
should be removed
SafeMath.sub(uint256,uint256,string) (goattoken.sol#1065-1074) is never used and
should be removed
SafeMath.tryAdd(uint256,uint256) (goattoken.sol#919-925) is never used and should be
removed
SafeMath.tryDiv(uint256,uint256) (goattoken.sol#961-966) is never used and should be
removed
SafeMath.tryMod(uint256,uint256) (goattoken.sol#973-978) is never used and should be
removed
SafeMath.tryMul(uint256,uint256) (goattoken.sol#944-954) is never used and should be
removed
SafeMath.trySub(uint256,uint256) (goattoken.sol#932-937) is never used and should be
removed
SafeMathInt.abs(int256) (goattoken.sol#2257-2260) is never used and should be
removed
SafeMathInt.div(int256,int256) (goattoken.sol#2228-2234) is never used and should be
removed
SafeMathInt.mul(int256,int256) (goattoken.sol#2216-2223) is never used and should be
removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Low level call in Address.sendValue(address,uint256) (goattoken.sol#558-563):
▌- (success) = recipient.call{value: amount}() (goattoken.sol#561)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string)
(goattoken.sol#626-637):
▌- (success,returndata) = target.call{value: value}(data) (goattoken.sol#635)
Low level call in Address.functionStaticCall(address,bytes,string)
(goattoken.sol#655-664):
▌- (success,returndata) = target.staticcall(data) (goattoken.sol#662)
Low level call in Address.functionDelegateCall(address,bytes,string)
(goattoken.sol#682-691):
▌- (success,returndata) = target.delegatecall(data) (goattoken.sol#689)
Low level call in SafeERC20NoRevert.safeTransfer(IERC20,address,uint256)
(goattoken.sol#1226-1238):
▌- (success,returndata) =
address(token).call(abi.encodeWithSelector(token.transfer.selector,to,value))
(goattoken.sol#1231-1233)
```

```
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Router01.WETH() (goattoken.sol#1284) is not in mixedCase
Function ContextUpgradeable.__Context_init() (goattoken.sol#1683-1685) is not in mixedCase
Function ContextUpgradeable.__Context_init_unchained() (goattoken.sol#1687-1688) is not in mixedCase
Variable ContextUpgradeable.__gap (goattoken.sol#1696) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init(string,string) (goattoken.sol#1754-1757) is not in mixedCase
Function ERC20Upgradeable.__ERC20_init_unchained(string,string) (goattoken.sol#1759-1762) is not in mixedCase
Variable ERC20Upgradeable.__gap (goattoken.sol#2061) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init() (goattoken.sol#2093-2096) is not in mixedCase
Function OwnableUpgradeable.__Ownable_init_unchained() (goattoken.sol#2098-2100) is not in mixedCase
Variable OwnableUpgradeable.__gap (goattoken.sol#2142) is not in mixedCase
Function IUniswapV2Pair.DOMAIN_SEPARATOR() (goattoken.sol#2165) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (goattoken.sol#2166) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (goattoken.sol#2183) is not in mixedCase
Function DividendPayingToken.__DividendPayingToken_init(address,string,string) (goattoken.sol#2475-2483) is not in mixedCase
Parameter DividendPayingToken.__DividendPayingToken_init(address,string,string)._rewardToken (goattoken.sol#2476) is not in mixedCase
Parameter DividendPayingToken.__DividendPayingToken_init(address,string,string)._name (goattoken.sol#2477) is not in mixedCase
Parameter DividendPayingToken.__DividendPayingToken_init(address,string,string)._symbol (goattoken.sol#2478) is not in mixedCase
Parameter DividendPayingToken.dividendOf(address)._owner (goattoken.sol#2538) is not in mixedCase
Parameter DividendPayingToken.withdrawableDividendOf(address)._owner (goattoken.sol#2545) is not in mixedCase
```

```
Parameter DividendPayingToken.withdrawnDividendOf(address)._owner
(goattoken.sol#2557) is not in mixedCase
Parameter DividendPayingToken.accumulativeDividendOf(address)._owner
(goattoken.sol#2571) is not in mixedCase
Constant DividendPayingToken.magnitude (goattoken.sol#2455) is not in
UPPER_CASE_WITH_UNDERSCORES
Parameter BABYTOKENDividendTracker.getAccount(address)._account (goattoken.sol#2742)
is not in mixedCase
Parameter AntiBotBABYTOKEN.setEnableAntiBot(bool)._enable (goattoken.sol#3097) is
not in mixedCase
Variable AntiBotBABYTOKEN._marketingWalletAddress (goattoken.sol#2979) is not in
mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-
to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint
256,address,uint256).amountADesired (goattoken.sol#1289) is too similar to IUniswapV2
Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256
).amountBDesired (goattoken.sol#1290)
Variable
DividendPayingToken.__DividendPayingToken_init(address,string,string)._rewardToken
(goattoken.sol#2476) is too similar to
BABYTOKENDividendTracker.initialize(address,uint256).rewardToken_
(goattoken.sol#2669)
Variable DividendPayingToken._withdrawDividendOfUser(address)._withdrawableDividend
(goattoken.sol#2510) is too similar to
BABYTOKENDividendTracker.getAccount(address).withdrawableDividends
(goattoken.sol#2749)
Variable ERC20._totalSupply (goattoken.sol#185) is too similar to AntiBotBABYTOKEN.co
nstructor(string,string,uint256,address[5],uint256[3],uint256,address,uint256).totalS
upply_ (goattoken.sol#3023)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-
names-are-too-similar

Clones.clone(address) (goattoken.sol#1151-1160) uses literals with too many digits:
▌- mstore(uint256,uint256)
(ptr_clone_asm_0,0x3d602d80600a3d3981f3363d3d373d3d3d363d730000000000000000000000000000)
(goattoken.sol#1154)
```

```
Clones.clone(address) (goattoken.sol#1151-1160) uses literals with too many digits:
▌- mstore(uint256,uint256)(ptr_clone_asm_0 +
0x28,0x5af43d82803e903d91602b57fd5bf3000000000000000000000000000000000000)
(goattoken.sol#1156)
Clones.cloneDeterministic(address,bytes32) (goattoken.sol#1169-1178) uses literals
with too many digits:
▌- mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0,0x3d602d80600a3d3981f3363d3d3
73d3d3d363d7300000000000000000000000000) (goattoken.sol#1172)
Clones.cloneDeterministic(address,bytes32) (goattoken.sol#1169-1178) uses literals
with too many digits:
▌- mstore(uint256,uint256)(ptr_cloneDeterministic_asm_0 +
0x28,0x5af43d82803e903d91602b57fd5bf3000000000000000000000000000000000000)
(goattoken.sol#1174)
Clones.predictDeterministicAddress(address,bytes32,address)
(goattoken.sol#1183-1198) uses literals with too many digits:
▌- mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0,0x3d602d80600a3d3981
f3363d3d373d3d3d363d7300000000000000000000000000) (goattoken.sol#1190)
Clones.predictDeterministicAddress(address,bytes32,address)
(goattoken.sol#1183-1198) uses literals with too many digits:
▌- mstore(uint256,uint256)(ptr_predictDeterministicAddress_asm_0 +
0x28,0x5af43d82803e903d91602b57fd5bf3ff0000000000000000000000000000000000)
(goattoken.sol#1192)
AntiBotBABYTOKEN.constructor(string,string,uint256,address[5],uint256[3],uint256,addr
ess,uint256) (goattoken.sol#3020-3095) uses literals with too many digits:
▌- gasForProcessing = 300000 (goattoken.sol#3053)
AntiBotBABYTOKEN.updateGasForProcessing(uint256) (goattoken.sol#3173-3184) uses
literals with too many digits:
▌- require(bool,string)(newValue >= 200000 && newValue <= 500000,BABYTOKEN:
gasForProcessing must be between 200,000 and 500,000) (goattoken.sol#3174-3177)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-
digits

SafeMathInt.MAX_INT256 (goattoken.sol#2211) is never used in SafeMathInt
(goattoken.sol#2209-2266)
OwnableUpgradeable.__gap (goattoken.sol#2142) is never used in
BABYTOKENDividendTracker (goattoken.sol#2644-2908)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-
```

```
state-variable

name() should be declared external:
▌- ERC20.name() (goattoken.sol#207-209)
symbol() should be declared external:
▌- ERC20.symbol() (goattoken.sol#215-217)
decimals() should be declared external:
▌- ERC20.decimals() (goattoken.sol#232-234)
transfer(address,uint256) should be declared external:
▌- ERC20.transfer(address,uint256) (goattoken.sol#258-261)
allowance(address,address) should be declared external:
▌- ERC20.allowance(address,address) (goattoken.sol#266-268)
approve(address,uint256) should be declared external:
▌- ERC20.approve(address,uint256) (goattoken.sol#277-280)
transferFrom(address,address,uint256) should be declared external:
▌- ERC20.transferFrom(address,address,uint256) (goattoken.sol#295-309)
increaseAllowance(address,uint256) should be declared external:
▌- ERC20.increaseAllowance(address,uint256) (goattoken.sol#323-326)
decreaseAllowance(address,uint256) should be declared external:
▌- ERC20.decreaseAllowance(address,uint256) (goattoken.sol#342-350)
renounceOwnership() should be declared external:
▌- Ownable.renounceOwnership() (goattoken.sol#877-879)
transferOwnership(address) should be declared external:
▌- Ownable.transferOwnership(address) (goattoken.sol#885-888)
name() should be declared external:
▌- ERC20Upgradeable.name() (goattoken.sol#1767-1769)
symbol() should be declared external:
▌- ERC20Upgradeable.symbol() (goattoken.sol#1775-1777)
decimals() should be declared external:
▌- ERC20Upgradeable.decimals() (goattoken.sol#1792-1794)
transfer(address,uint256) should be declared external:
▌- ERC20Upgradeable.transfer(address,uint256) (goattoken.sol#1818-1821)
allowance(address,address) should be declared external:
▌- ERC20Upgradeable.allowance(address,address) (goattoken.sol#1826-1828)
approve(address,uint256) should be declared external:
▌- ERC20Upgradeable.approve(address,uint256) (goattoken.sol#1837-1840)
transferFrom(address,address,uint256) should be declared external:
```

```
▌- ERC20Upgradeable.transferFrom(address,address,uint256) (goattoken.sol#1855-1869)
increaseAllowance(address,uint256) should be declared external:
▌- ERC20Upgradeable.increaseAllowance(address,uint256) (goattoken.sol#1883-1886)
decreaseAllowance(address,uint256) should be declared external:
▌- ERC20Upgradeable.decreaseAllowance(address,uint256) (goattoken.sol#1902-1910)
renounceOwnership() should be declared external:
▌- OwnableUpgradeable.renounceOwnership() (goattoken.sol#2124-2126)
transferOwnership(address) should be declared external:
▌- OwnableUpgradeable.transferOwnership(address) (goattoken.sol#2132-2135)
get(IterableMapping.Map,address) should be declared external:
▌- IterableMapping.get(IterableMapping.Map,address) (goattoken.sol#2299-2301)
getIndexOfKey(IterableMapping.Map,address) should be declared external:
▌- IterableMapping.getIndexOfKey(IterableMapping.Map,address)
(goattoken.sol#2303-2312)
getKeyAtIndex(IterableMapping.Map,uint256) should be declared external:
▌- IterableMapping.getKeyAtIndex(IterableMapping.Map,uint256)
(goattoken.sol#2314-2320)
size(IterableMapping.Map) should be declared external:
▌- IterableMapping.size(IterableMapping.Map) (goattoken.sol#2322-2324)
distributeCAKEDividends(uint256) should be declared external:
▌- DividendPayingToken.distributeCAKEDividends(uint256) (goattoken.sol#2485-2496)
withdrawDividend() should be declared external:
▌- BABYTOKENDividendTracker.withdrawDividend() (goattoken.sol#2689-2694)
▌- DividendPayingToken.withdrawDividend() (goattoken.sol#2500-2502)
dividendOf(address) should be declared external:
▌- DividendPayingToken.dividendOf(address) (goattoken.sol#2538-2540)
withdrawnDividendOf(address) should be declared external:
▌- DividendPayingToken.withdrawnDividendOf(address) (goattoken.sol#2557-2564)
isExcludedFromDividends(address) should be declared external:
▌- BABYTOKENDividendTracker.isExcludedFromDividends(address)
(goattoken.sol#2706-2712)
getAccountAtIndex(uint256) should be declared external:
▌- BABYTOKENDividendTracker.getAccountAtIndex(uint256) (goattoken.sol#2791-2812)
process(uint256) should be declared external:
▌- BABYTOKENDividendTracker.process(uint256) (goattoken.sol#2839-2891)
updateGasForProcessing(uint256) should be declared external:
▌- AntiBotBABYTOKEN.updateGasForProcessing(uint256) (goattoken.sol#3173-3184)
```

```
isExcludedFromFees(address) should be declared external:
▌- AntiBotBABYTOKEN.isExcludedFromFees(address) (goattoken.sol#3213-3215)
withdrawableDividendOf(address) should be declared external:
▌- AntiBotBABYTOKEN.withdrawableDividendOf(address) (goattoken.sol#3217-3223)
dividendTokenBalanceOf(address) should be declared external:
▌- AntiBotBABYTOKEN.dividendTokenBalanceOf(address) (goattoken.sol#3225-3231)
isExcludedFromDividends(address) should be declared external:
▌- AntiBotBABYTOKEN.isExcludedFromDividends(address) (goattoken.sol#3237-3243)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-
function-that-could-be-declared-external
goattoken.sol analyzed (30 contracts with 78 detectors), 162 result(s) found
```

RUGDOG

WOOF!

rugdog.net

the@rugdog.net