

SKELETON ECOSYSTEM

SMART CONTRACT AUDIT



DEPT. OF GOVERNMENT EFFICIENCY

D.O.G.E.

BEP20

0x0632365f723CDb003F125A94254A4d9DCcd25051



Table of Contents

Table of Contents	1
Disclaimer	2
Overview	3
Creation/Audit Date	3
Verified Socials	3
Contract Functions Analysis	4
Contract Safety and Weakness	9
Detected Vulnerability Description	13
Contract Flow Graph	16
Contract Interaction Graph	17
Inheritance Graph	18
Contract Descriptions	19
Audit Scope	30

Global Disclaimer

This document serves as a disclaimer for the crypto smart contract audit conducted by Skeleton Ecosystem. The purpose of the audit was to review the codebase of the smart contracts for potential vulnerabilities and issues. It is important to note the following:

Limited Scope: The audit is based on the code and information available up to the audit completion date. It does not cover external factors, system interactions, or changes made after the audit. The audit itself can not guarantee 100% safety and can not detect common scam methods like farming and developer sell-out.

No Guarantee of Security: While we have taken reasonable steps to identify vulnerabilities, it is impossible to guarantee the complete absence of security risks or issues. The audit report provides an assessment of the contract's security as of the audit date.

Continued Development: Smart contracts and blockchain technology are evolving fields. Updates, forks, or changes to the contract post-audit may introduce new risks that were not present during the audit.

Third-party Code: If the smart contract relies on third-party libraries or code, those components were not thoroughly audited unless explicitly stated. Security of these dependencies is the responsibility of their respective developers.

Non-Exhaustive Testing: The audit involved automated analysis, manual review, and testing under controlled conditions. It is possible that certain vulnerabilities or issues may not have been identified.

Risk Evaluation: The audit report includes a risk assessment for identified vulnerabilities. It is recommended that the development team carefully reviews and addresses these risks to mitigate potential exploits.

Not Financial Advice: This audit report is not intended as financial or investment advice. Decisions regarding the use, deployment, or investment in the smart contract should be made based on a comprehensive assessment of the associated risks.

By accessing and using this audit report, you acknowledge and agree to the limitations outlined above. Skeleton Ecosystem and its auditors shall not be held liable for any direct or indirect damages resulting from the use of the audit report or the smart contract itself.

Please consult with legal, technical, and financial professionals before making any decisions related to the smart contract.

Overview

Contract Name	\$DOGE
Ticker/Symbol	\$D.O.G.E.
Blockchain	Binance Smart Chain BEP20
Contract Address	0x0632365f723CDb003F125A94254A4d9DCcd25051
Creator Address	0x3886989d984236D60296268befd1eF0feCf2CE13
Current Owner Address	0x00000000000000000000000000000000dEaD
Contract Explorer	https://bscscan.com/address/0x0632365f723cdb003f125a94254a4d9dccc25051#code
Compiler Version	v0.8.15+commit.e14f2714
License	MIT
Optimisation	Yes with 200 Runs
Total Supply	100,000 \$D.O.G.E
Decimals	18

Creation/Audit

Contract Deployed	23.10.2024
Audit Created	15.01.2025
Audit Update	V 1.0

Verified Socials

Website	https://dogeonbsc.net/
Telegram	https://t.me/dogeddept_bsc
Twitter (X)	https://x.com/DOGEBSC_ARMY

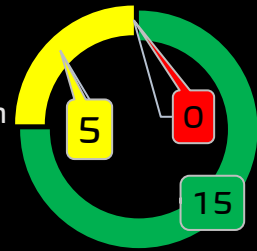
Contract Function Analysis








 Pass
  Attention Item
  Risky Item











■ Pass

■ Attention

■ Risk






Contract Verified		The contract source code is uploaded to blockchain explorer and is open source, so everybody can read it.
Contract Ownership		0x00000000000000000000000000000000dEaD
Buy Tax	3 %	Shows the taxes for purchase transactions. Above 10% may be considered a high tax rate. More than 50% tax rate means may not be tradable. Fee can be set!
Sell Tax	3 %	Shows the taxes for sell transactions. Above 10% may be considered a high tax rate. More than 50% tax rate means may not be tradable. Fee can be set!
Honeypot Analyse		Holder is able to buy and sell. If honeypot: The contract blocks sell transfer from holder wallet. Multiple events may cause honeypot. Trading disabled, extremely high tax
Liquidity Status		Liquidity status on 15.01.2025 Lp Locked: 100% on Pinksale locker for 282 days. https://bscscan.com/tx/0x2fbb26f0575d0860223cdb5427334e48639c96af08b770f599ddacc2ac5b4ab0
Trading Disable Functions		No Trading suspendable function found. If a suspendable code is included, the token maybe neither be bought or sold (honeypot risk). If contract is renounced this function can't be used
Set Fees function	 35%max	Fee Setting function found. Max 35% Contract Renounced. Function can not be triggered The contract owner may contain the authority to modify the transaction tax. If the transaction tax is increased to more than 49%, the tokens may not be able to be traded (honeypot risk).
Proxy Contract		Not a Proxy contract
Mint Function		No Mint Function detected Mint function is transparent or non-existent. Hidden mint functions may increase the amount of tokens in circulation and effect the price of the token. Owner can mint new tokens and sell.

Balance Modifier Function		<p>No Balance Modifier function found.</p> <p>If there is a function for this, the contract owner can have the authority to modify the balance of tokens at other addresses. For example revoke the bought tokens from the holders wallet. Common form of scam: You buy the token, but it's disappearing from your wallet.</p>
Blacklist Function		<p>No Blacklist Setting function found.</p> <p>If there is a blacklist, some addresses may not be able to trade normally. Example: you buy the token and right after your Wallet getting blacklisted. Like so you will be unable to sell. Honeypot Risk.</p>
Whitelist Function		<p>Whitelist Setting function found</p> <p>Contract Renounced. Function can not be triggered</p> <p>If there is a function for this Developer can set zero fee or no max wallet size for addresses (for example team wallets can trade without fee. Can cause farming)</p>
Hidden Owner Analysis		<p>No Hidden or multi owner with authorisation</p> <p>For contract with a hidden owner, developer can still manipulate the contract even if the ownership has been abandoned.</p>
Retrieve Ownership Function		<p>No Functions found which can retrieve ownership of the contract.</p> <p>If this function exists, it is possible for the project owner to regain ownership even after relinquishing it. Also known as fake renounce.</p>
Self Destruct Function		<p>No Self Destruct function found.</p> <p>If this function exists and is triggered, the contract will be destroyed, all functions will be unavailable, and all related assets will be erased.</p>
Transfer Tax Changing Function	 max15%	<p>Transfer Tax Changing Functions found.</p> <p>Contract Renounced. Function can not be triggered</p> <p>If it exists, the contract owner may set a very outrageous tax rate for assigned address to block it from trading. Can assign all wallets at once!</p>
Trading Cooldown Function		<p>Trading Cooldown Function found. If there is a trading cooldown function, the user will not be able to sell the token within a certain time or block after buying. Like a temporary honeypot.</p>
Max Transaction and Holding Modify Function		<p>Max Transaction and Holding Modify function found.</p> <p>Contract Renounced. Function can not be triggered</p> <p>If there is a function for this, the maximum trading amount or maximum position can be modified. Can cause honeypot</p>
Transaction Limiting Function		<p>No Transaction Limiter Function Found.</p> <p>The number of overall token transactions may be limited (honeypot risk)</p>

Details of Risk - Attention Items

Following detected contract functions serve as informational purposes about the contract. The owner has no more authorisation to trigger the following functions.

Address	0x0632365f723cdb003f125a94254a4d9dccc25051 (Dept. Of Government Efficiency: \$D.O.G.E Token)   
Name	OwnershipTransferred (index_topic_1 address previousOwner, index_topic_2 address newOwner) View Source
Topics	<div>0 0x8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e0</div> <div>1: previousOwner Dec ▾ → 0x3886989d984236D60296268befd1eF0feCf2CE13</div> <div>2: newOwner Dec ▾ → 0x00</div>
Data	0x

Set Fee [35% max found]

Contract Renounced. Function can not be triggered

The contract owner may contain the authority to modify the transaction tax. If the transaction tax is increased to more than 49%, the tokens may not be able to be traded (honeypot risk).

```

1278  function updateFees(
1279      uint256 deadBuy!,
1280      uint256 deadSell!,
1281      uint256 marketingBuy!,
1282      uint256 marketingSell!,
1283      uint256 liquidityBuy!,
1284      uint256 liquiditySell!,
1285      uint256 RewardsBuy!,
1286      uint256 RewardsSell!,
1287      uint256 devBuy!,
1288      uint256 devSell!
1289  ) public onlyOwner {
1290      buyDeadFees = deadBuy!;
1291      buyMarketingFees = marketingBuy!;
1292      buyLiquidityFee = liquidityBuy!;
1293      buyRewardsFee = RewardsBuy!;
1294      sellDeadFees = deadSell!;
1295      sellMarketingFees = marketingSell!;
1296      sellLiquidityFee = liquiditySell!;
1297      sellRewardsFee = RewardsSell!;
1298      buyDevFee = devBuy!;
1299      sellDevFee = devSell!;
1300
1301      totalSellFees = sellRewardsFee
1302          .add(sellLiquidityFee)
1303          .add(sellMarketingFees)
1304          .add(sellDevFee);
1305
1306      totalBuyFees = buyRewardsFee
1307          .add(buyLiquidityFee)
1308          .add(buyMarketingFees)
1309          .add(buyDevFee);
1310
1311      require(totalSellFees <= 35 && totalBuyFees <= 35, "total fees cannot exceed 35% sell or buy");
1312  }
  
```

⚠️ Whitelist

If there is a function for this Developer can set zero fee or no max wallet size for addresses (for example team wallets can trade without fee. Can cause farming)

Contract Renounced. Function can not be triggered

```

1151
1152 // exclude a wallet from fees
1153 ftrace | funcSig
1154 function setExcludeFees(address account!, bool excluded!) public onlyOwner {
1155     | isExcludedFromFees[account!] = excluded!;
1156     | emit ExcludeFromFees(account!, excluded!);
1157 }
1158
1159 ftrace | funcSig
1339 function isExcludedFromFees(address account!) public view returns (bool) {
1340     | return isExcludedFromFees[account!];
1341 }
1342

```

⚠️ Max Transaction and Holding Modify function

If there is a function for this, the maximum trading amount or maximum position can be modified. Can cause honeypot

Contract Renounced. Function can not be triggered

```

1194 // set max wallet, can not be lower than 0.05% of supply
1195 ftrace | funcSig
1196 function setmaxWallet(uint256 value!) external onlyOwner {
1197     | value! = value! * (10**18);
1198     | require(value! >= _totalSupply / 2000, "max wallet cannot be set to less than 0.05%");
1199     | maxWallet = value!;
1200 }

```

⚠️ Transfer Tax update

Transfer between wallets can be taxed, resulting holding loss when sending AVE token from wallet A to B

Contract Renounced. Function can not be triggered

```

1272 ftrace | funcSig
1273 function updateTransferFee(uint256 newTransferFee!) public onlyOwner {
1274     | require (newTransferFee! <= 15, "transfer fee cannot exceed 15%");
1275     | transferFee = newTransferFee!;
1276     | emit UpdateTransferFee(transferFee);
1277 }

```


⚠️ Cooldown

Trading Cooldown Function found. If there is a trading cooldown function, the user will not be able to sell the token within a certain time or block after buying. Like a temporary honeypot.

Contract Renounced. Function can not be triggered

```
1187
1188     // set cooldown timer, can only be between 0 and 300 seconds (5 mins max)
1189     ftrace | funcSig
1189     function setcooldowntimer(uint256 value) external onlyOwner {
1190         require(value <= 300, "cooldown timer cannot exceed 5 minutes");
1191         cooldowntimer = value;
1192     }
1193
```

Total Findings: 7

■ High 0

■ Medium 0

■ Low 4

■ Info 2



■ **High Severity Issues:** High possibility to cause problems, need to be resolved.

■ **Medium Severity Issue:** Will likely cause problems, recommended to resolve.

■ **Low Severity Issues:** Won't cause problems, but for improvement purposes could be adjusted.

■ **Informational Severity Issues:** Not harmful in any way, information for the developer team.

Contract Security

List of Found Issues

 **High severity Issues: (0)**

 **Medium severity issues: (0)**

 **Low severity issues: (4)**

- Missing Events
- Long number literals
- Outdated compiler Version
- Approve of front running attack

 **Informational severity issues: (2)**

- Public Functions Should be Declared External
- State Variables Should be Declared Constant

Contract Weakness Classification

THE SMART CONTRACT WEAKNESS CLASSIFICATION REGISTRY (SWC REGISTRY) IS AN IMPLEMENTATION OF THE WEAKNESS CLASSIFICATION SCHEME PROPOSED IN EIP-1470. IT IS LOOSELY ALIGNED TO THE TERMINOLOGIES AND STRUCTURE USED IN THE COMMON WEAKNESS ENUMERATION (CWE) WHILE OVERLAYING A WIDE RANGE OF WEAKNESS VARIANTS THAT ARE

ID	Description	AI	Manual	Result
SWC-100	Function Default Visibility	Passed	Passed	Passed
SWC-101	Integer Overflow and Underflow	Passed	Passed	Passed
SWC-102	Outdated Compiler Version	low	low	low
SWC-103	Floating Pragma	low	Passed	Passed
SWC-104	Unchecked Call Return Value	Passed	Passed	Passed
SWC-105	Unprotected Ether Withdrawal	Passed	Passed	Passed
SWC-106	Unprotected SELFDESTRUCT Instruction	Passed	Passed	Passed
SWC-107	Reentrancy	Passed	Passed	Passed
SWC-108	State Variable Default Visibility	Passed	Passed	Passed
SWC-109	Uninitialized Storage Pointer	Passed	Passed	Passed
SWC-110	Assert Violation	Passed	Passed	Passed
SWC-111	Use of Deprecated Solidity Functions	Passed	Passed	Passed
SWC-112	Delegatecall to Untrusted Callee	Passed	Passed	Passed
SWC-113	DoS with Failed Call	Passed	Passed	Passed
SWC-114	Transaction Order Dependence	Passed	Passed	Passed
SWC-115	Authorization through tx.origin	Passed	Passed	Passed
SWC-116	Block values as a proxy for time	Passed	Passed	Passed
SWC-117	Signature Malleability	Passed	Passed	Passed
SWC-118	Incorrect Constructor Name	Passed	Passed	Passed
SWC-119	Shadowing State Variables	Passed	Passed	Passed
SWC-120	Weak Sources of Randomness from Chain Attributes	Passed	Passed	Passed

SWC-121	Missing Protection against Signature Replay Attacks	Passed	Passed	Passed
SWC-122	Lack of Proper Signature Verification	Passed	Passed	Passed
SWC-123	Requirement Violation	Passed	Passed	Passed
SWC-124	Write to Arbitrary Storage Location	Passed	Passed	Passed
SWC-125	Incorrect Inheritance Order	Passed	Passed	Passed
SWC-126	Insufficient Gas Griefing	Passed	Passed	Passed
SWC-127	Arbitrary Jump with Function Type Variable	Passed	Passed	Passed
SWC-128	DoS With Block Gas Limit	Passed	Passed	Passed
SWC-129	Typographical Error	low	Passed	Passed
SWC-130	Right-To-Left-Override control character (U+202E)	Passed	Passed	Passed
SWC-131	Presence of unused variables	Passed	Passed	Passed
SWC-132	Unexpected Ether balance	Passed	Passed	Passed
SWC-133	Hash Collisions With Multiple Variable Length Arguments	Passed	Passed	Passed
SWC-134	Message call with hardcoded gas amount	Passed	Passed	Passed
SWC-135	Code With No Effects	Passed	Passed	Passed
SWC-136	Unencrypted Private Data On-Chain	Passed	Passed	Passed

Detected High and Medium Severity Vulnerability Description.

⚠ Approve of Front running Attack (2 Item)

Sandwich bot attack

Item: 1	Location:	Line 265-273	Severity:	Low
---------	-----------	--------------	-----------	-----

Function	<p>The <code>approve()</code> method overrides current allowance regardless of whether the spender already used it or not, so there is no way to increase or decrease allowance by a certain value atomically unless the token owner is a smart contract, not an account. This can be abused by a token receiver when they try to withdraw certain tokens from the sender's account. Meanwhile, if the sender decides to change the amount and sends another approve transaction, the receiver can notice this transaction before it's mined and can extract tokens from both the transactions, therefore, ending up with tokens from both the transactions. This is a front-running attack affecting the ERC20 Approve function.</p> <p>The function approve can be front-run by abusing the <code>_approve</code> function.</p>
Remediation	<ol style="list-style-type: none"> 1. Introduce mechanisms that limit the maximum acceptable gas price for transactions. This can help prevent front-runners from drastically increasing the gas fees to prioritize their transactions. 2. Use transaction taxes to prevent against front-run attack

```

265 | ftrace | funcSig
266 | function approve(address spender, uint256 amount)
267 |     public
268 |     virtual
269 |     override
270 |     returns (bool)
271 | {
272 |     _approve(_msgSender(), spender, amount);
273 |     return true;
274 | }
  
```

Item: 2	Location:	Line 275-290	Severity:	Low
---------	-----------	--------------	-----------	-----

Function	<p>The transferFrom() method overrides current allowance regardless of whether the spender already used it or not, so there is no way to increase or decrease allowance by a certain value atomically unless the token owner is a smart contract, not an account.</p> <p>This can be abused by a token receiver when they try to withdraw certain tokens from the sender's account.</p> <p>Meanwhile, if the sender decides to change the amount and sends another approve transaction, the receiver can notice this transaction before it's mined and can extract tokens from both the transactions, therefore, ending up with tokens from both the transactions. This is a front-running attack affecting the ERC20 Approve function.</p> <p>The function transferFrom can be front-run by abusing the <code>_approve</code> function.</p>
Remedation	<ol style="list-style-type: none"> 1. Introduce mechanisms that limit the maximum acceptable gas price for transactions. This can help prevent front-runners from drastically increasing the gas fees to prioritize their transactions. 2. Use transaction taxes to prevent against front-run attack

```

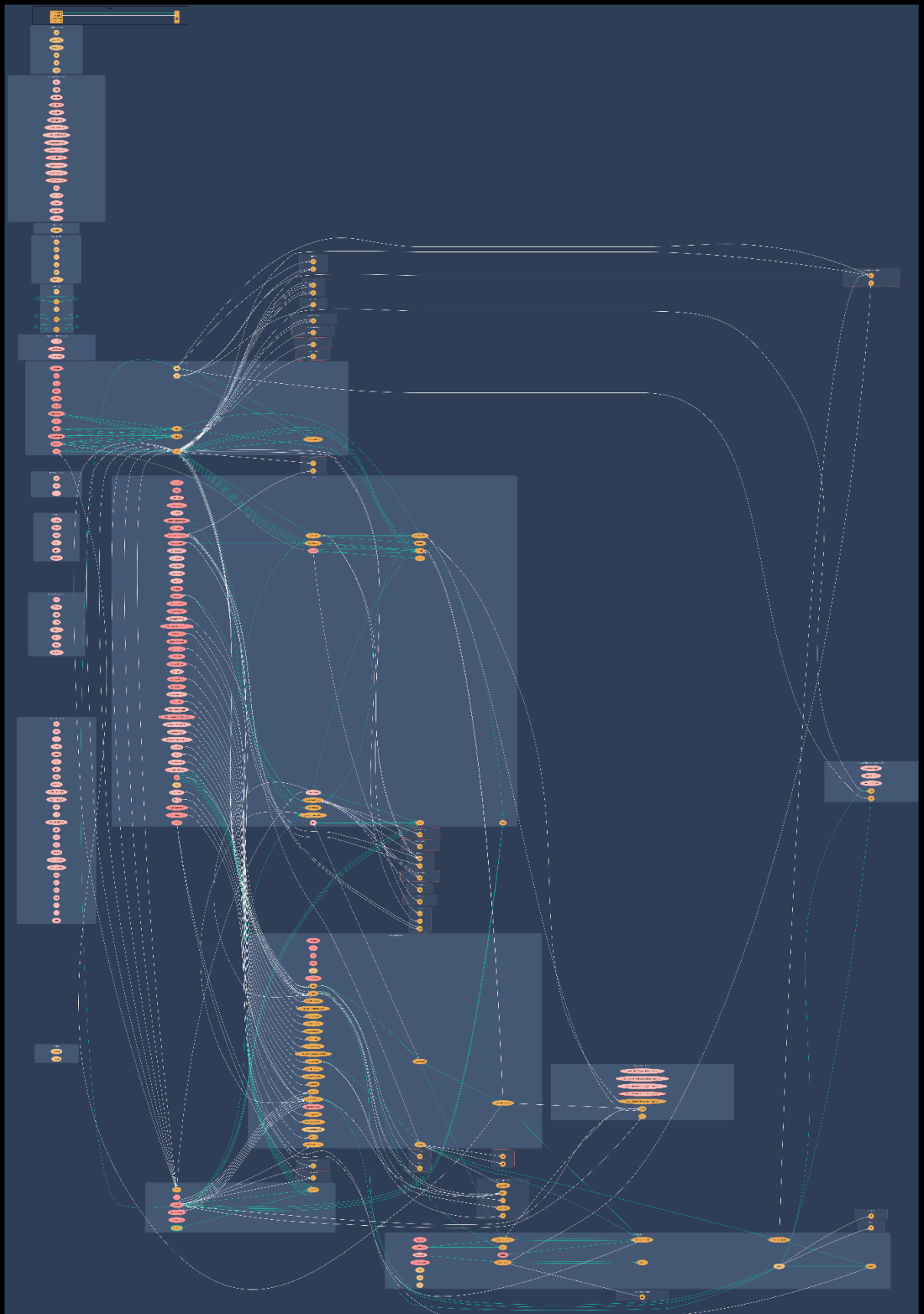
275  ftrace | funcSig
276  function transferFrom(
277      address sender!,
278      address recipient!,
279      uint256 amount!
280  ) public virtual override returns (bool) {
281      _transfer(sender!, recipient!, amount!);
282      _approve(
283          sender!,
284          _msgSender(),
285          allowances[sender!][_msgSender()].sub(
286              amount!,
287              "ERC20: transfer amount exceeds allowance"
288          );
289      return true;
290  }
291
  
```

Outdated Compiler Version.

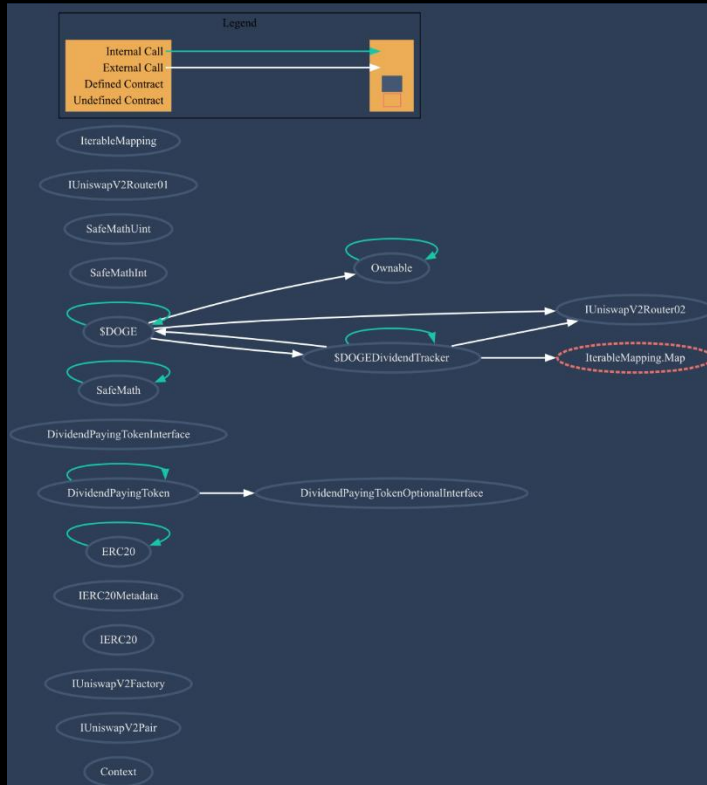
Item: 1	Location:	Line 11	Severity:	 Low
---------	-----------	---------	-----------	---

Function	Using an outdated compiler version can be problematic especially if there are publicly disclosed bugs and issues that affect the current compiler version. The following outdated versions were detected: /doge.sol - ^0.8.15
Remedation	It is recommended to use a recent version of the Solidity compiler that should not be the most recent version, and it should not be an outdated version as well. Using very old versions of Solidity prevents the benefits of bug fixes and newer security checks. Consider using the solidity version v0.8.26, which patches most solidity vulnerabilities.

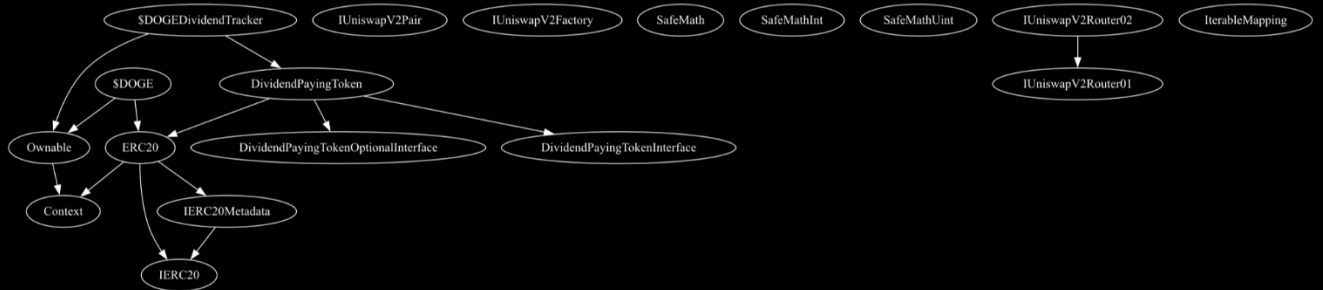
Contract Flow Graph











































Contract Interaction Graph
















Inheritance Graph












































Contract Functions











Contract	Type	Bases		
L	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
L	_msgSender	Internal 		
L	_msgData	Internal 		
IUniswapV2Pair	Interface			
L	name	External 		NO 
L	symbol	External 		NO 
L	decimals	External 		NO 
L	totalSupply	External 		NO 
L	balanceOf	External 		NO 
L	allowance	External 		NO 
L	approve	External 		NO 
L	transfer	External 		NO 
L	transferFrom	External 		NO 
L	DOMAIN_SEPARATOR	External 		NO 
L	PERMIT_TYPEHASH	External 		NO 
L	nonces	External 		NO 
L	permit	External 		NO 
L	MINIMUM_LIQUIDITY	External 		NO 
L	factory	External 		NO 
L	token0	External 		NO 
L	token1	External 		NO 















































L	getReserves	External ¶		NO ¶
L	price0CumulativeLast	External ¶		NO ¶
L	price1CumulativeLast	External ¶		NO ¶
L	kLast	External ¶		NO ¶
L	mint	External ¶	●	NO ¶
L	burn	External ¶	●	NO ¶
L	swap	External ¶	●	NO ¶
L	skim	External ¶	●	NO ¶
L	sync	External ¶	●	NO ¶
L	initialize	External ¶	●	NO ¶
IUniswapV2Factory	Interface			
L	feeTo	External ¶		NO ¶
L	feeToSetter	External ¶		NO ¶
L	getPair	External ¶		NO ¶
L	allPairs	External ¶		NO ¶
L	allPairsLength	External ¶		NO ¶
L	createPair	External ¶	●	NO ¶
L	setFeeTo	External ¶	●	NO ¶
L	setFeeToSetter	External ¶	●	NO ¶
IERC20	Interface			
L	totalSupply	External ¶		NO ¶
L	balanceOf	External ¶		NO ¶
L	transfer	External ¶	●	NO ¶
L	allowance	External ¶		NO ¶

L	approve	External ⓘ		NO ⓘ
L	transferFrom	External ⓘ		NO ⓘ
IERC20Metadata	Interface	IERC20		
L	name	External ⓘ		NO ⓘ
L	symbol	External ⓘ		NO ⓘ
L	decimals	External ⓘ		NO ⓘ
ERC20	Implementation	Context, IERC20, IERC20Metadata		
L		Public ⓘ		NO ⓘ
L	name	Public ⓘ		NO ⓘ
L	symbol	Public ⓘ		NO ⓘ
L	decimals	Public ⓘ		NO ⓘ
L	totalSupply	Public ⓘ		NO ⓘ
L	balanceOf	Public ⓘ		NO ⓘ
L	transfer	Public ⓘ		NO ⓘ
L	allowance	Public ⓘ		NO ⓘ
L	approve	Public ⓘ		NO ⓘ
L	transferFrom	Public ⓘ		NO ⓘ
L	increaseAllowance	Public ⓘ		NO ⓘ
L	decreaseAllowance	Public ⓘ		NO ⓘ
L	_transfer	Internal 🔒		
L	_mint	Internal 🔒		
L	_burn	Internal 🔒		
L	_approve	Internal 🔒		
L	_beforeTokenTransfer	Internal 🔒		














































DividendPayingTo kenOptionalInterf ace	Interface			
L	withdrawableDivi dendOf	External ¶		NO ¶
L	withdrawnDividen dOf	External ¶		NO ¶
L	accumulativeDivid endOf	External ¶		NO ¶
DividendPayingTo kenInterface	Interface			
L	dividendOf	External ¶		NO ¶
L	distributeDividend s	External ¶	Ⓢ	NO ¶
L	withdrawDividend	External ¶	Ⓢ	NO ¶
SafeMath	Library			
L	add	Internal 🔒		
L	sub	Internal 🔒		
L	sub	Internal 🔒		
L	mul	Internal 🔒		
L	div	Internal 🔒		
L	div	Internal 🔒		
L	mod	Internal 🔒		
L	mod	Internal 🔒		
Ownable	Implementation	Context		
L		Public ¶	Ⓢ	NO ¶
L	owner	Public ¶		NO ¶
L	renounceOwnersh ip	Public ¶	Ⓢ	onlyOwner
L	transferOwnersh ip	Public ¶	Ⓢ	onlyOwner








SafeMathInt	Library			
L	mul	Internal 		
L	div	Internal 		
L	sub	Internal 		
L	add	Internal 		
L	abs	Internal 		
L	toUint256Safe	Internal 		
SafeMathUint	Library			
L	toInt256Safe	Internal 		
IUniswapV2Router01	Interface			
L	factory	External 		NO 
L	WETH	External 		NO 
L	addLiquidity	External 		NO 
L	addLiquidityETH	External 		NO 
L	removeLiquidity	External 		NO 
L	removeLiquidityETH	External 		NO 
L	removeLiquidityETHWithPermit	External 		NO 
L	removeLiquidityETHWithPermit	External 		NO 
L	swapExactTokensForTokens	External 		NO 
L	swapTokensForExactTokens	External 		NO 
L	swapExactETHForTokens	External 		NO 
L	swapTokensForExactETH	External 		NO 
































L	swapExactTokensForETH	External ¶		NO ¶
L	swapETHForExactTokens	External ¶		NO ¶
L	quote	External ¶		NO ¶
L	getAmountOut	External ¶		NO ¶
L	getAmountIn	External ¶		NO ¶
L	getAmountsOut	External ¶		NO ¶
L	getAmountsIn	External ¶		NO ¶
IUniswapV2Router02	Interface	IUniswapV2Router01		
L	removeLiquidityETHSupportingFeeOnTransferTokens	External ¶		NO ¶
L	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ¶		NO ¶
L	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ¶		NO ¶
L	swapExactETHForTokensSupportingFeeOnTransferTokens	External ¶		NO ¶
L	swapExactTokensForETHSupportingFeeOnTransferTokens	External ¶		NO ¶
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
L		Public ¶		ERC20
L		External ¶		NO ¶
L	distributeDividends	Public ¶		NO ¶

L	withdrawDividend	Public 		NO 
L	_withdrawDividendOfUser	Internal 		
L	dividendOf	Public 		NO 
L	withdrawableDividendOf	Public 		NO 
L	withdrawnDividendOf	Public 		NO 
L	accumulativeDividendOf	Public 		NO 
L	_transfer	Internal 		
L	_mint	Internal 		
L	_burn	Internal 		
L	_setBalance	Internal 		
\$DOGE	Implementation	ERC20, Ownable		
L		Public 		ERC20
L	decimals	Public 		NO 
L		External 		NO 
L	updateStakingAmounts	Public 		onlyOwner
L	enableTrading	External 		onlyOwner
L	setPresaleWallet	External 		onlyOwner
L	setExcludeFees	Public 		onlyOwner
L	setExcludeDividends	Public 		onlyOwner
L	setIncludeDividends	Public 		onlyOwner
L	setCanTransferBefore	External 		onlyOwner
L	setLimitsInEffect	External 		onlyOwner
L	setGasPriceLimit	External 		onlyOwner

L	setcooldowntimer	External ⓘ		onlyOwner
L	setMaxWallet	External ⓘ		onlyOwner
L	enableStaking	Public ⓘ		onlyOwner
L	stake	Public ⓘ		NO ⓘ
L	setSwapTriggerAmount	Public ⓘ		onlyOwner
L	enableSwapAndLiquify	Public ⓘ		onlyOwner
L	setAutomatedMarketMakerPair	Public ⓘ		onlyOwner
L	setAllowCustomTokens	Public ⓘ		onlyOwner
L	setAllowAutoReinvest	Public ⓘ		onlyOwner
L	_setAutomatedMarketMakerPair	Private 🔒		
L	updateGasForProcessing	Public ⓘ		onlyOwner
L	transferAdmin	Public ⓘ		onlyOwner
L	updateTransferFee	Public ⓘ		onlyOwner
L	updateFees	Public ⓘ		onlyOwner
L	getStakingInfo	External ⓘ		NO ⓘ
L	getTotalDividendsDistributed	External ⓘ		NO ⓘ
L	isExcludedFromFees	Public ⓘ		NO ⓘ
L	withdrawableDividendOf	Public ⓘ		NO ⓘ
L	dividendTokenBalanceOf	Public ⓘ		NO ⓘ
L	getAccountDividendsInfo	External ⓘ		NO ⓘ
L	getAccountDividendsInfoAtIndex	External ⓘ		NO ⓘ

L	processDividendTracker	External 		NO 
L	claim	External 		NO 
L	getLastProcessedIndex	External 		NO 
L	getNumberOfDividendTokenHolders	External 		NO 
L	setAutoClaim	External 		NO 
L	setReinvest	External 		NO 
L	setDividendsPaused	External 		onlyOwner
L	isExcludedFromAutoClaim	External 		NO 
L	isReinvest	External 		NO 
L	_transfer	Internal 		
L	getStakingBalance	Private 		
L	swapAndLiquify	Private 		
L	swapTokensForEth	Private 		
L	updatePayoutToken	Public 		onlyOwner
L	getPayoutToken	Public 		NO 
L	setMinimumTokenBalanceForAutoDividends	Public 		onlyOwner
L	setMinimumTokenBalanceForDividends	Public 		onlyOwner
L	addLiquidity	Private 		
L	forceSwapAndSendDividends	Public 		onlyOwner
L	swapAndSendDividends	Private 		
L	multiSend	Public 		onlyOwner

L	airdropToWallets	External 		onlyOwner
\$DOGEDividendTracker	Implementation	DividendPayingToken, Ownable		
L		Public 		DividendPayingToken
L	decimals	Public 		NO 
L	name	Public 		NO 
L	symbol	Public 		NO 
L	_transfer	Internal 		
L	withdrawDividend	Public 		NO 
L	isExcludedFromAutoClaim	External 		onlyOwner
L	isReinvest	External 		onlyOwner
L	setAllowCustomTokens	External 		onlyOwner
L	setAllowAutoReinvest	External 		onlyOwner
L	excludeFromDividends	External 		onlyOwner
L	includeFromDividends	External 		onlyOwner
L	setAutoClaim	External 		onlyOwner
L	setReinvest	External 		onlyOwner
L	setMinimumTokenBalanceForAutoDividends	External 		onlyOwner
L	setMinimumTokenBalanceForDividends	External 		onlyOwner
L	setDividendsPaused	External 		onlyOwner
L	getLastProcessedIndex	External 		NO 

L	getNumberOfTokenHolders	External 		NO 
L	getAccount	Public 		NO 
L	getAccountAtIndex	Public 		NO 
L	setBalance	External 		onlyOwner
L	process	Public 		NO 
L	processAccount	Public 		onlyOwner
L	updateUniswapV2Router	Public 		onlyOwner
L	updatePayoutToken	Public 		onlyOwner
L	getPayoutToken	Public 		NO 
L	_reinvestDividendOfUser	Private 		
L	_withdrawDividendOfUser	Internal 		
IterableMapping	Library			
L	get	Internal 		
L	getIndexOfKey	Internal 		
L	getKeyAtIndex	Internal 		
L	size	Internal 		
L	set	Internal 		
L	remove	Internal 		



Function
can modify
state



Function
is payable

Audit Scope

Audit Method.

Our smart contract audit is an extensive methodical examination and analysis of the smart contract's code that is used to interact with the blockchain. Goal: discover errors, issues and security vulnerabilities in the code. Findings getting reported and improvements getting suggested.

Automatic and Manual Review

We are using automated tools to scan functions and weaknesses of the contract. Transfers, integer over-undeflow checks such as all CWE events.

Tools we use:

Visual Studio Code

CWE

SWC

Solidity Scan

SVD

In manual code review our auditor looking at source code and performing line by line examination. This method helps to clarify developer's coding decisions and business logic.

Skeleton Ecosystem

<https://skeletonecosystem.com>

<https://github.com/SkeletonEcosystem/Audits>

