Cyberscope

Audit Report

Black Doge

August 2022

Type	BEP20
Network	BSC
Address	0xbdfF32474E3B2fDe9Ff306B9a8BEffbc3c3DC0Fd
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	BlackDoge
Compiler Version	v0.8.3+commit.8d00100c
Optimization	200 runs
Licence	GNU GPLv3
Explorer	https://bscscan.com/token/0xbdfF32474E3B2fDe9Ff306B9a8BEffbc3c3DC0Fd
Symbol	BLDOGE
Decimals	18
Total Supply	100,000,000,000,000
Domain	blackdoge.app

Source Files

Filename	**SHA256**
contract.sol	1c6864a4c75b754bf19ab2687b303f56ca09f93467347399691130b88e37398d

Audit Updates

Initial Audit	3rd August 2022
Corrected	

Contract Analysis

Critical Medium Minor Pass

Severity	Code	Description
●	ST	Contract Owner is not able to stop or pause transactions
●	OCTD	Contract Owner is not able to transfer tokens from specific address
●	OTUT	Owner Transfer User's Tokens
●	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
●	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
●	MT	Contract Owner is not able to mint new tokens
●	BT	Contract Owner is not able to burn tokens from specific wallet
●	BC	Contract Owner is not able to blacklist wallets from selling

ST - Stop Transactions

Criticality	minor
Location	contract.sol#L988,1025

Description

The contract could potentially revert the sales. Once the contract finishes a ‘swapBack()’ operation, the ‘_liquidityTokensToSwap, _buyBackTokensToSwap and _marketingTokensToSwap’ variables will reset to zero. If someone deposits tokens to the contract, then the contract will trigger the ‘swapBack()’ in the next sale transaction but the ‘totalTokensToSwap’ will be zero. As a result, the contract will execute a zero division calculation and the transaction will revert.

The contract’s state will recover once a non-sale transfer takes place. Therefore, the ‘totalTokensToSwap’ will contain a non-zero value.

```
        if (
            !inSwapAndLiquify &&
            swapAndLiquifyEnabled &&
            balanceOf(uniswapV2Pair) > 0
        ) {
            if (automatedMarketMakerPairs[to]) {
                if (
                    overMinimumTokenBalance
                ) {
                    swapBack();

function swapBack() private lockTheSwap {
        uint256 contractBalance = balanceOf(address(this));
        uint256 totalTokensToSwap =
_liquidityTokensToSwap.add(_buyBackTokensToSwap).add(_marketingTokensToSwap);
        // Halve the amount of liquidity tokens
        uint256 tokensForLiquidity = _liquidityTokensToSwap.div(2);
        uint256 amountToSwapForBNB = contractBalance.sub(tokensForLiquidity);
        uint256 initialBNBBalance = address(this).balance;
        swapTokensForBNB(amountToSwapForBNB);
        uint256 bnbBalance = address(this).balance.sub(initialBNBBalance);
        uint256 bnbForOperations =
bnbBalance.mul(_marketingTokensToSwap).div(totalTokensToSwap);
```

Recommendation

The contract should embody a check for allowing `swapBack` only if `totalTokensToSwap` is greater than zero.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	STC	Succeeded Transfer Check
	CO	Code Optimization
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L05	Unused State Variable
	L07	Missing Events Arithmetic
	L09	Dead Code Elimination
	L13	Divide before Multiply Operation

STC - Succeeded Transfer Check

Criticality	minor
Location	contract.sol#L1398

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
function transferToAddressETH(address payable recipient, uint256 amount)
private
{
    recipient.transfer(amount);
}
```

Recommendation

The contract should check if the result of the transfer methods is successful.

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1398

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations.

This code segment is not used on the contract's implementation. As a result it is redundant.

```
function transferToAddressETH(address payable recipient, uint256 amount)
private
{
    recipient.transfer(amount);
}
```

Recommendation

Rewrite some code segments so the runtime will be more performant.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L855,230,1389,248,235,776,244

Description

Public functions that are never called by the contract should be declared external to save gas.

```
getUnlockTime
approve
transferOwnership
getTime
setSwapAndLiquifyEnabled
renounceOwnership
setAutomatedMarketMakerPair
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L651,680,207,645,208

Description

Constant state variables should be declared constant to save gas.

```
_lockTime
_marketingFee
_previousOwner
minimumTokensBeforeSwap
_buybackFee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L1431,660,1305,1366,1385,322,352,662,636,1376,320,1309,1404,654,656,632,1389,655,398,1381,1371,657,659,661,637,638

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_decimals
_symbol
_sellMarketingFee
_sellTaxFee
_buyBuybackFee
_buybackAddress
_teamAddress
WETH
_buyLiquidityFee
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L652,207,646

Description

There are segments that contain unused state variables.

```
_previousMarketingFee
_previousOwner
_previousBuyBackFee
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1344,1355,876,871

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
gasPriceLimit = gas * 1000000000
gasMaxLimit = gas * 1000000000
_sellTaxFee = sellTaxFee
_buyTaxFee = buyTaxFee
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L123,152,166,1394,179,113,144,137

Description

Functions that are not used in the contract, and make the code's size bigger.

```
functionCall
isContract
_functionCallWithValue
transferToAddressETH
functionCallWithValue
sendValue
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1021

Description

Performing divisions before multiplications may cause lose of prediction.

```
bnbForOperations = bnbBalance.mul(_marketingTokensToSwap).div(totalTokensToSwap)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	getUnlockTime	Public		-
	getTime	Public		-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-

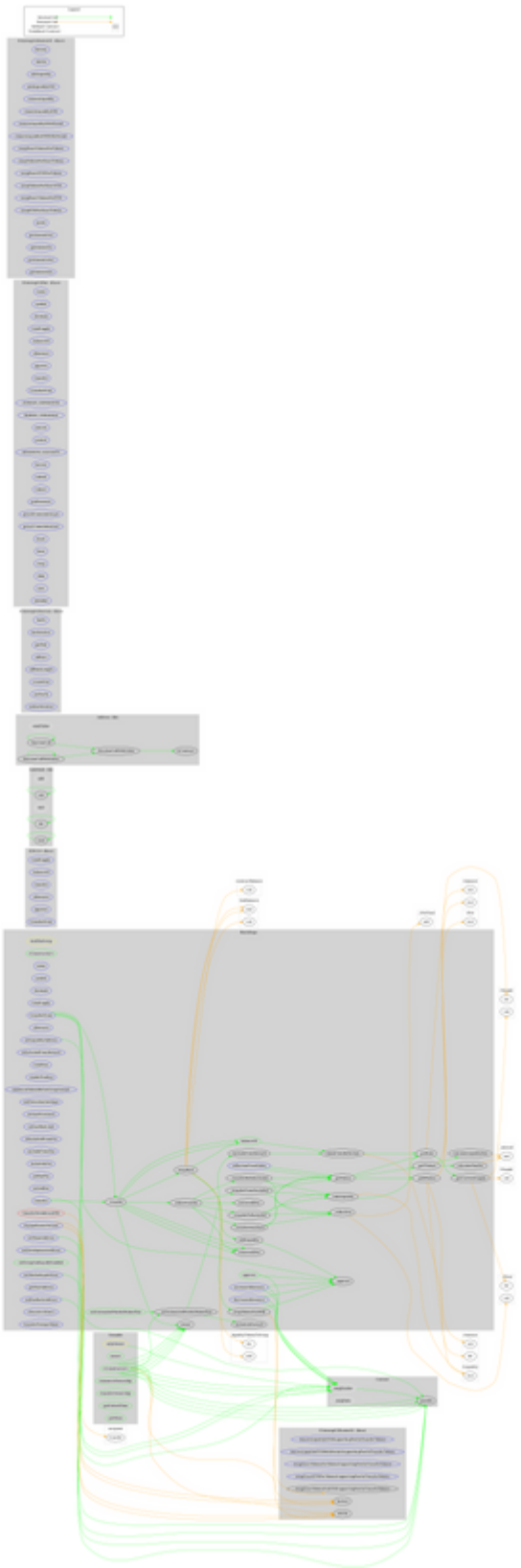
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-

IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
BlackDoge	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	External		-
	symbol	External		-
	decimals	External		-
	totalSupply	External		-
	balanceOf	Public		-
	transfer	External	✓	-
	allowance	External		-
	approve	Public	✓	-
	transferFrom	External	✓	-
	increaseAllowance	External	✓	-
	decreaseAllowance	External	✓	-
	isExcludedFromReward	External		-
	totalFees	External		-
	enableTrading	External	✓	onlyOwner
	minimumTokensBeforeSwapAmount	External		-
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	setProtectionSettings	External	✓	onlyOwner
	setGasPriceLimit	External	✓	onlyOwner
	setGasMaxLimit	External	✓	onlyOwner
	reflectionFromToken	External		-

	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	Public	✓	onlyOwner
	_approve	Private	✓	
	_transfer	Private	✓	
	swapBack	Private	✓	lockTheSwap
	swapTokensForBNB	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	External		-
	excludeFromFee	External	✓	onlyOwner
	includeInFee	External	✓	onlyOwner
	setBuyFee	External	✓	onlyOwner
	setSellFee	External	✓	onlyOwner
	setMarketingAddress	External	✓	onlyOwner
	setBuyBackAddress	External	✓	onlyOwner
	setLiquidityAddress	External	✓	onlyOwner
	setTeamAddress	External	✓	onlyOwner
	setDevelopmentAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner

	transferToAddressETH	Private	✓	
	getPairAddress	External		onlyOwner
	changeRouterVersion	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	transferForeignToken	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	blackdoge.app
Registry Domain ID	49AD3A58A-APP
Creation Date	2022-07-29T12:26:31Z
Updated Date	2022-08-03T12:44:44Z
Registry Expiry Date	2024-07-29T12:26:31Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	Namecheap Inc.
Registrar IANA ID	1068

The domain was created 2 years before the creation of the audit.

There is no public billing information, the creator is protected by the privacy settings.

Summary

Black Doge is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract has renounced the ownership, as a result the owner functions can not be used in a malicious way to disturb the users' transactions. The fees are fixed to 10% in buys and 15% in sales.

Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io