Cyberscope

Audit Report

THE BEND

March 2022

Table of Contents

Contract Review

Contract Name	BEND
Compiler Version	v0.8.0+commit.c7dfd78e
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x13AbEC309d958c7900e8DEA7d930D794981507Ad
Symbol	BEND
Decimals	9
Total Supply	1,000,000
Domain	thebend.io

Source Files

Filename	**SHA256**
contract.sol	85a7ba87a1506a1a2cb3974485684a17de20cb142c2ebcab94d057bbf4d0b8d6

Audit Updates

Initial Audit	26th March 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

ELFM - Exceed Limit Fees Manipulation

Criticality	medium
Location	contract.sol#L1748

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `updateFees` function with the maximum allowed values and have a total sell or buy fee of 50%.

```solidity
function updateFees(
    uint256 operationsSellTax,
    uint256 operationsBuyTax,
    uint256 liquidityTax,
    uint256 BENDRewardsBuyTax,
    uint256 BENDRewardsSellTax
) public onlyOwner {
    require(operationsSellTax <= 30, "operation fees can't exceed 30%");
    require(operationsBuyTax <= 30, "operation fees can't exceed 30%");
    require(liquidityTax <= 10, "liquidity fees can't exceed 10%");
    require(
        BENDRewardsBuyTax >= 0 && BENDRewardsBuyTax <= 10,
        "BENDreward fees must be between 0 and 10"
    );
    require(
        BENDRewardsSellTax >= 0 && BENDRewardsSellTax <= 10,
        "BEND reward fees must be between 0 and 10"
    );

    operationsSellFee = operationsSellTax;
    operationsBuyFee = operationsBuyTax;
    liquidityFee = liquidityTax;
    BENDRewardsBuyFee = BENDRewardsBuyTax;
    BENDRewardsSellFee = BENDRewardsSellTax;
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Staking Feature

Criticality	Informational
Location	contract.sol#L1917

Description

The Contract Implements a staking feature. The user has the ability to stake his tokens for a period of time. Within that period he will not be able to sell or transfer his tokens, and he will not be able to cancel the staking procedure until the period he selected is over. The user will get rewarded in terms of StakingBonus. That means that he will get increased dividend shares based on the duration of the stake period he has chosen.

```
if (!automatedMarketMakerPairs[from] && stakingEnabled) {
        require(
            stakingUntilDate[from] <= block.timestamp,
            "Tokens are staked and locked!"
        );
        if (stakingUntilDate[from] != 0) {
            stakingUntilDate[from] = 0;
            stakingBonus[from] = 0;
        }
    }
```

Contract Diagnostics

● Critical   ● Medium   ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L08	Tautology or Contradiction
●	L09	Dead Code Elimination
●	L12	Using Variables before Declaration
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope
●	L15	Local Scope Variable Shadowing

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L355,363,1570,412,425,442,465,494,521,907 and 36 more

Description

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

```
getAllowTokens
updateAllowTokens
getPayoutToken
updatePayoutToken
updateUniswapV2Router
process
getAccountAtIndex
forceSwapAndSendDividends
dividendTokenBalanceOf

...
```

Recommendation

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

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L48,50,81,996,1308,1315,1327,1341,1211,1581 and 6 more

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.

```
_account
BENDRewardsSellFee
BENDRewardsBuyFee
BENDRewardsSellTax
BENDRewardsBuyTax
_contractAddress
UpdatestakeAddress
magnitude
_owner
...
```

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#L928

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L1632,1643,2112

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.

```
buyAmount = buyAmount.sub(fromBuy)
swapTokensAtAmount = newNum * (10 ** 9)
maxSellTransactionAmount = newNum * (10 ** 9)
```

Recommendation

Emit an event for critical parameter changes.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L1748

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(BENDRewardsSellTax >= 0 && BENDRewardsSellTax <= 10,BEND
reward fees must be between 0 and 10)
require(bool,string)(BENDRewardsBuyTax >= 0 && BENDRewardsBuyTax <=
10,BENDreward fees must be between 0 and 10)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L1360,1276,163,974

Description

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

```
abs
get
_withdrawDividendOfUser
_transfer
```

Recommendation

Remove unused functions.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L2004,2003,2005

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
lastProcessedIndex
iterations
claims
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1893,2123

Description

Performing divisions before multiplications may cause lose of prediction.

```
dividendsFromSell =
address(this).balance.mul(sellAmount).div(totalAmount).mul(BENDRewardsSellFee).
div(BENDRewardsSellFee.add(operationsSellFee))
dividendsFromBuy =
address(this).balance.mul(buyAmount).div(totalAmount).mul(BENDRewardsBuyFee).di
v(BENDRewardsBuyFee.add(operationsBuyFee))
swapAmountSold = contractTokenBalance.mul(sellAmount).div(totalBuySell)
swapAmountBought = contractTokenBalance.mul(buyAmount).div(totalBuySell)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L2003,2004,2514,2577,2005

Description

The are variables that are defined in the local scope and are not initialized.

```
lastProcessedIndex
success
claims
iterations
```

Recommendation

All the local scoped variables should be initialized.

L15 - Local Scope Variable Shadowing

Criticality	minor
Location	contract.sol#L1231

Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_symbol
_name
```

Recommendation

The local variables should have different names from the upper scoped variables.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
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		-
	mint	External	✓	-
	burn	External	✓	-

	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IterableMapping	Library			
	get	Internal		
	getIndexOfKey	Internal		
	getKeyAtIndex	Internal		
	size	Internal		
	set	Internal	✓	
	remove	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-

	symbol	External		-
	decimals	External		-
ERC20	Implementation	Context, IERC20, IERC20Met adata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
DividendPayin gTokenOption alInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
DividendPayin gTokenInterfac e	Interface			
	dividendOf	External		-
	distributeDividends	External	Payable	-
	withdrawDividend	External	✓	-

SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Ownable	Implementation	Context		
	\<Constructor\>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
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	✓	-
DividendPayingToken	Implementation	ERC20, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	distributeDividends	Public	Payable	-
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	

	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
BEND	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	decimals	Public		-
	recoveryBalance	External	✓	onlyOwner
	UpdatestakeAddress	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	updateStakingAmounts	Public	✓	onlyOwner
	setPresaleWallet	External	✓	onlyOwner
	enableStaking	Public	✓	onlyOwner
	stake	Public	✓	-
	updateMaxAmount	Public	✓	onlyOwner
	updateSwapTokensAtAmount	Public	✓	onlyOwner
	updateDividendTracker	Public	✓	onlyOwner
	updateOperations1Address	Public	✓	onlyOwner
	updateOperations2Address	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	enableSwapAndLiquify	Public	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	setAllowCustomTokens	Public	✓	onlyOwner
	setAllowAutoReinvest	Public	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateLiquidityWallet	Public	✓	onlyOwner
	updateGasForProcessing	Public	✓	onlyOwner
	updateFees	Public	✓	onlyOwner
	getStakingInfo	External		-

	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-
	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	setAutoClaim	External	✓	-
	setReinvest	External	✓	-
	setDividendsPaused	External	✓	onlyOwner
	isExcludedFromAutoClaim	External		-
	isReinvest	External		-
	_transfer	Internal	✓	
	getStakingBalance	Private		
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	updatePayoutToken	Public	✓	-
	getPayoutToken	Public		-
	updateAllowTokens	Public	✓	onlyOwner
	getAllowTokens	Public		-
	addLiquidity	Private	✓	
	forceSwapAndSendDividends	Public	✓	onlyOwner
	swapAndSendDividends	Private	✓	
BENDDividend Tracker	Implementation	DividendPayingToken, Ownable		
	<Constructor>	Public	✓	DividendPayingToken
	decimals	Public		-
	_transfer	Internal		
	withdrawDividend	Public		-
	isExcludedFromAutoClaim	External		onlyOwner

	isReinvest	External		onlyOwner
	setAllowCustomTokens	External	✓	onlyOwner
	setAllowAutoReinvest	External	✓	onlyOwner
	excludeFromDividends	External	✓	onlyOwner
	setAutoClaim	External	✓	onlyOwner
	setReinvest	External	✓	onlyOwner
	setDividendsPaused	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	updatePayoutToken	Public	✓	onlyOwner
	getPayoutToken	Public		-
	updateAllowTokens	Public	✓	onlyOwner
	getAllowTokens	Public		-
	_reinvestDividendOfUser	Private	✓	
	_withdrawDividendOfUser	Internal	✓	

Contract Flow
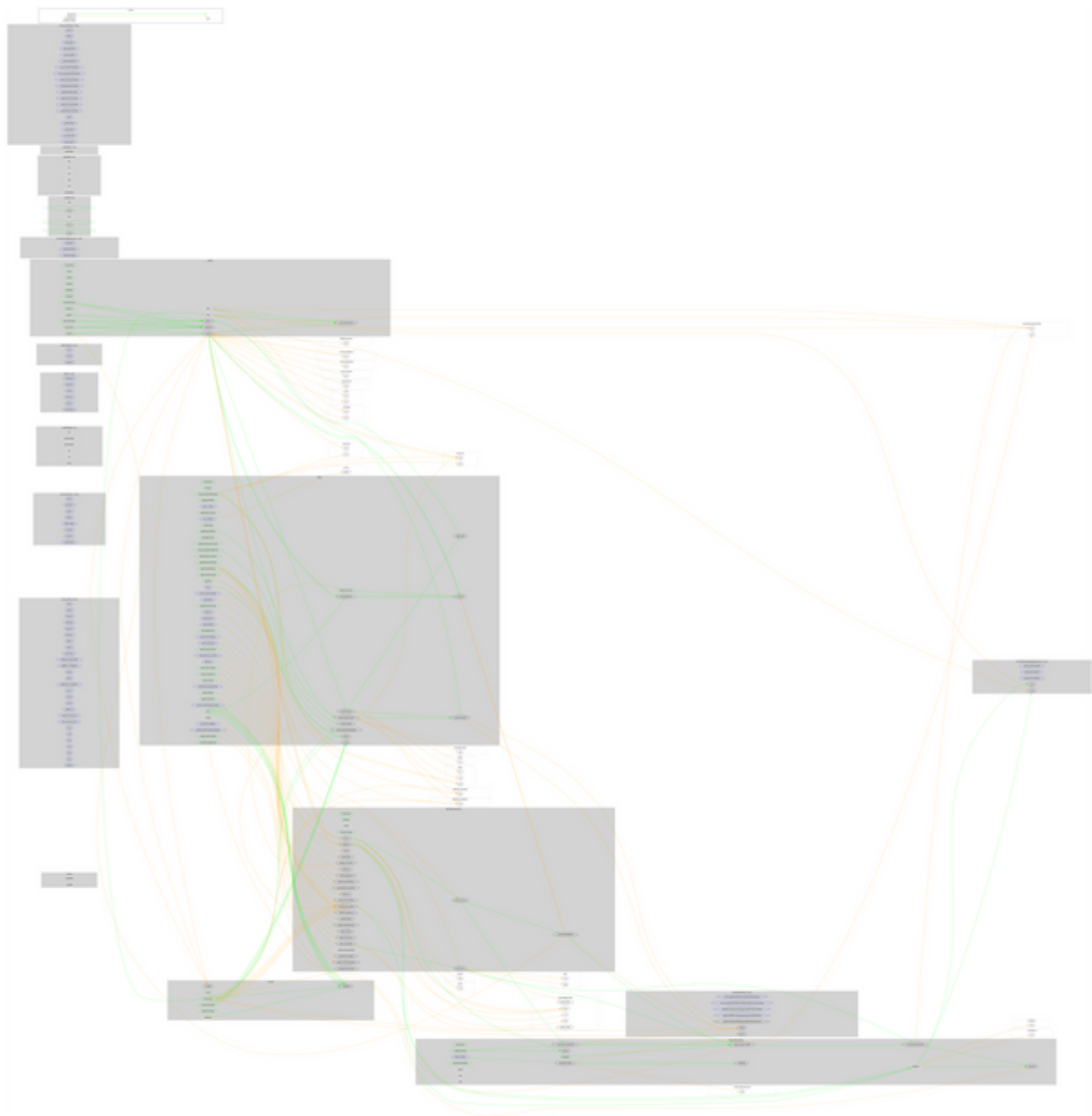
Domain Info

Domain Name	thebend.io
Registry Domain ID	807526baab7a4536ae026df04dd3fbc5-DONUTS
Creation Date	2022-02-24T21:44:50Z
Updated Date	2022-03-08T11:55:09Z
Registry Expiry Date	2023-02-24T21:44:50Z
Registrar WHOIS Server	whois.godaddy.com/
Registrar URL	http://www.godaddy.com/domains/search.aspx?ci=8990
Registrar	GoDaddy.com, LLC
Registrar IANA ID	146

The domain has been created 30 days before the creation of the audit. It will expire in 11 months.

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

Summary

The Bend is an interesting project with a friendly and growing community. The analysis reported one medium threat issue. The contract owner can change the fees up to 50%. The contract also implements a staking feature that allows the user to temporarily lock his tokens, and in return he will earn more dividends as a Staking Bonus. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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 provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io