Cyberscope

Audit Report

DinoX Inu

November 2022

Type	BEP20
Network	BSC
Address	0x3eEE2f5eCAf39140B40D2911A73239f41B8A732D
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	DinoXInuContract
Compiler Version	v0.8.15+commit.e14f2714
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x3eEE2f5eCAf39140B40D2911A73239f41B8A732D
Symbol	DOXI
Decimals	18
Total Supply	10,000,000,000,000
Domain	dinoinu.site

Source Files

Filename	SHA256
contract.sol	80b732ecff0e64dbcdcd504507ea276a20cb3834e8a10226ddc6f95e8a319713

Audit Updates

Initial Audit	13th November 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Contract Diagnostics

● Critical ● Medium ● Minor / Informative

Severity	Code	Description	Status
●	ZD	Zero Division	Unresolved
●	TSD	Total Supply Diversion	Unresolved
●	CO	Code Optimization	Unresolved
●	L02	State Variables could be Declared Constant	Unresolved
●	L04	Conformance to Solidity Naming Conventions	Unresolved
●	L09	Dead Code Elimination	Unresolved
●	L15	Local Scope Variable Shadowing	Unresolved

ZD - Zero Division

Criticality	critical
Location	contract.sol#L737
Status	Unresolved

Description

The contract is using variables that may be set to zero as denominators. As a result, the transactions will revert. This may happen if someone transfers $DOXI tokens to the contract right after a `swapAndLiquify()` event.

```
uint256 walletsTotal = devTokensCollected + marketingTokensCollected +
charityTokensCollected + reserveTokensCollected;

uint256 ethForMarketing = (newBalance * marketingTokensCollected)/walletsTotal;
```

Recommendation

The contract should prevent those variables to be set to zero or should not allow executing the corresponding statements.

TSD - Total Supply Diversion

Criticality	critical
Location	contract.sol#L737
Status	Unresolved

Description

The `manualBurn()` method transfers the caller's amount to the dead wallet. Additionally, it decreases the amount from the total supply. As a result, the total supply is diverse from the sum of balances.

```
function manualBurn(uint256 burnAmount) external isAuth
{
    removeAllFee();
        _tTotal = _tTotal - burnAmount;
    _transferStandard(owner(), deadWallet, burnAmount);
    restoreAllFee();
    emit Log("We have manually burned a Total Of:", burnAmount);

}
```

Recommendation

The sum of balances should always equal to the total supply.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L912
Status	Unresolved

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. The contract calculates the `rLiquidity` twice, once in the `_getRValues()` and once in the `_takeLiquidity()` method.

```
function _getRValues(uint256 tAmount, uint256 tFee, uint256 tLiquidity, uint256
currentRate) private pure returns (uint256, uint256, uint256) {
    uint256 rAmount = tAmount * currentRate;
    uint256 rFee = tFee * currentRate;
    uint256 rLiquidity = tLiquidity * currentRate;
    uint256 rTransferAmount = (rAmount - rFee) - rLiquidity;
    return (rAmount, rTransferAmount, rFee);
}
...

function _takeLiquidity(uint256 tLiquidity) private {
    uint256 currentRate =  _getRate();
    uint256 rLiquidity = tLiquidity * currentRate;
    _rOwned[address(this)] = _rOwned[address(this)] + rLiquidity;
    if(_isExcluded[address(this)]) { _tOwned[address(this)] =
_tOwned[address(this)] + tLiquidity; } //TODO: Verify Change
}
```

Recommendation

The contract could rewrite some code segments so the runtime will be more performant.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L470,469,467,466,468,465
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
_saleCharityFee
_saleMarketingFee
_saleReserveFee
_saleLiquidityFee
_saleDevFee
_saleTaxFee
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L978,961,445,442,972,917,467,440,470,441,416,235,251,460,469,921,454,457,234,451,465,966,951,1009,985,466,272,992,448,468
Status	Unresolved

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.

```
_devWallet
_minimumTokensBeforeSwap
_taxFee
_decimals
_charityWallet
_amount
_saleReserveFee
_name
_saleCharityFee
...
```

Recommendation

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

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L95,112,1012,116,121,84,108,104
Status	Unresolved

Description

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

```
sendValue
functionCallWithValue
swapETHForTokens
_functionCallWithValue
isContract
functionCall
```

Recommendation

Remove unused functions.

L15 - Local Scope Variable Shadowing

Criticality	minor / informative
Location	contract.sol#L519
Status	Unresolved

Description

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

```
_owner
```

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		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
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		-
	lock	Public	✓	onlyOwner

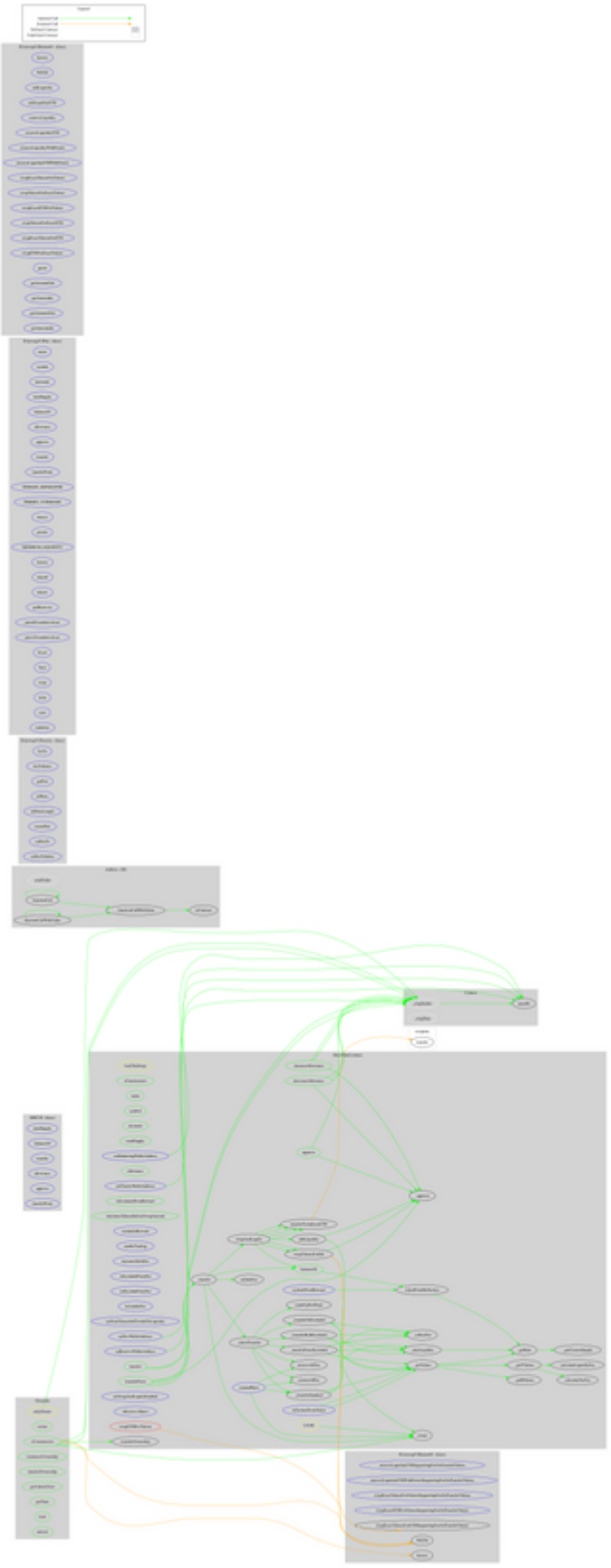
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	Interface			
	name	External		-
	symbol	External		-
	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	✓	-
	swapExactETHForTokensSupportingF	External	Payable	-

	eeOnTransferTokens			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
DinoXInuContract	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	minimumTokensBeforeSwapAmount	Public		-
	reflectionFromToken	External		-
	tokenFromReflection	Public		-
	excludeFromReward	External	✓	isAuth
	includeInReward	External	✓	isAuth
	_approve	Private	✓	
	enableTrading	External	✓	isAuth
	renounceSafuDev	External	✓	-
	_transfer	Private	✓	
	swapAndLiquify	Public	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	setSaleFee	Private	✓	
	countUpFeeShare	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		-
	setExcludeFromFee	External	✓	isAuth
	includeInFee	External	✓	isAuth
	setNumTokensSellToAddToLiquidity	External	✓	isAuth
	setMarketingWalletAddress	External	✓	onlyOwner
	setCharityWalletAddress	External	✓	onlyOwner
	setDevWalletAddress	External	✓	onlyOwner
	setReserveWalletAddress	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	External	✓	isAuth
	transferToAddressETH	Private	✓	
	<Receive Ether>	External	Payable	-
	swapETHForTokens	Private	✓	
	manualBurn	External	✓	isAuth

Contract Flow



Domain Info

Domain Name	dinoinu.site
Registry Domain ID	D320536576-CNIC
Creation Date	2022-09-01T16:03:52+00:00
Updated Date	2022-09-06T16:16:32+00:00
Registry Expiry Date	2023-09-01T23:59:59+00:00
Registrar WHOIS Server	whois.hostinger.com
Registrar URL	https://www.hostinger.com/
Registrar	Hostinger, UAB
Registrar IANA ID	1636

The domain was created 2 months before the creation of the audit. It will expire in 10 months.

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

Summary

DinoX Inu is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. The fees are fixed at 11%.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s 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 regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document 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 represents an extensive assessment 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.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io