Cyberscope

Audit Report

Diferencial

June 2022

Type	BEP20
Network	BSC
Address	0x027CF94720849C86e1f6A301aC610CBF4324f90A
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Diferencial
Compiler Version	v0.8.14+commit.80d49f37
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x027CF94720849C86e1f6A301aC610CBF4324f90A
Symbol	DRL
Decimals	9
Total Supply	100,000,000,000,000,000
Domain	diferencialtoken.com

Source Files

Filename	**SHA256**
contract.sol	5d292d02f5e1536fe87cf94e31783ea47d98d2ea5de0911bce1f6d357c5e4fd6

Audit Updates

Initial Audit	6th June 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	critical
Location	contract.sol#L354

Description

The contract owner has the authority to set an external contract to determine if the transfer should proceed. The external contract is not verified, so it works as an untrusted source. If the external contract contains vulnerabilities then the token can be converted into a honeypot and prevent users from selling.

```
function setInitializer(address initializer) external onlyOwner {
    require(!tradingEnabled);
    require(initializer != address(this), "Can't be self.");
    antiSnipe = AntiSnipe(initializer);
}
```

Recommendation

The contract could have a verified source pointed at it so the implementation would be readable and auditable. 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.

Contract Diagnostics

Critical Medium Minor

Severity	Code	Description
	L01	Public Function could be Declared External
	L02	State Variables could be Declared Constant
	L04	Conformance to Solidity Naming Conventions
	L07	Missing Events Arithmetic
	L11	Unnecessary Boolean equality
	L12	Using Variables before Declaration
	L13	Divide before Multiply Operation
	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L293,298,311,324,386,414,435,439,615,625,653

Description

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

```
isExcludedFromReward
enableTrading
getCirculatingSupply
getMaxWallet
getMaxTX
setRatios
isBlacklisted
setNewRouter
approveContractContingency
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L127,197

Description

Constant state variables should be declared constant to save gas.

```
totalReflections
_tTotal
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L33,394,121,123,124,125,129,146,152,161,162,163,164,165,176,192

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.

```
_hasLiqBeenAdded
_taxWallets
masterTaxDivisor
maxRoundtripTax
maxTransferTaxes
maxSellTaxes
maxBuyTaxes
_ratios
_taxRates
...
```

Recommendation

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

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L425,430,443,449,484

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.

```
burnLimit = (_tTotal * percent) / divisor
piSwapPercent = priceImpactSwapPercent
swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor
_maxWalletSize = (_tTotal * percent) / divisor
_maxTxAmount = (_tTotal * percent) / divisor
```

Recommendation

Emit an event for critical parameter changes.

L11 - Unnecessary Boolean equality

Criticality	minor
Location	contract.sol#L337

Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
enabled == false
```

Recommendation

Remove the equality to the boolean constant.

L12 - Using Variables before Declaration

Criticality	minor
Location	contract.sol#L754

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.

```
check
```

Recommendation

The variables should be declared before any usage of them.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L548,745

Description

Performing divisions before multiplications may cause lose of prediction.

```
feeAmount = (tAmount * currentFee) / masterTaxDivisor
toLiquify = ((contractTokenBalance * ratios.liquidity) / ratios.totalSwap) / 2
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L753,746,754

Description

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

```
check
values
checked
```

Recommendation

All the local scoped variables should be initialized.

Contract Functions

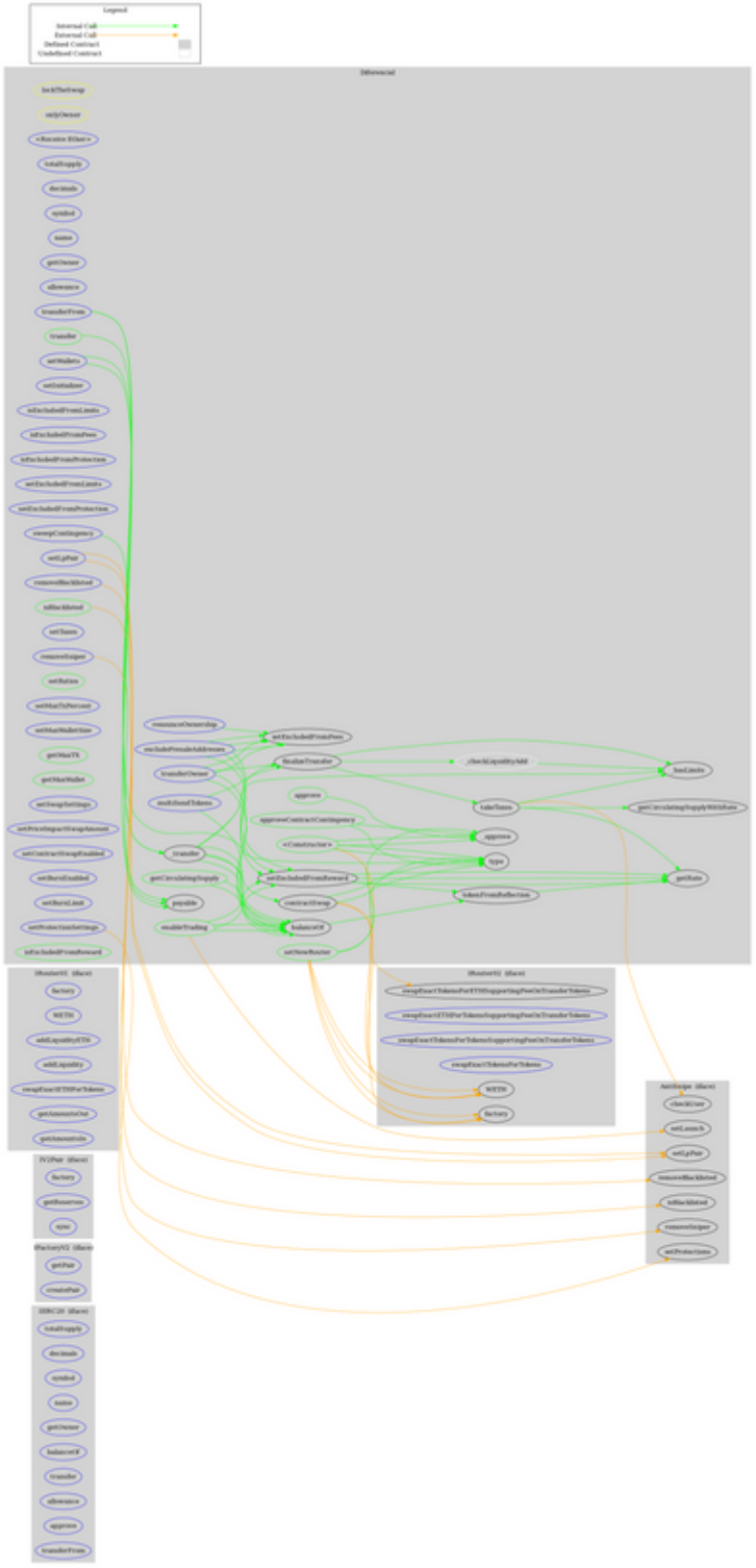
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IFactoryV2	Interface			
	getPair	External		-
	createPair	External	✓	-
IV2Pair	Interface			
	factory	External		-
	getReserves	External		-
	sync	External	✓	-
IRouter01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	addLiquidity	External	✓	-
	swapExactETHForTokens	External	Payable	-
	getAmountsOut	External		-

	getAmountsIn	External		-
IRouter02	Interface	IRouter01		
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokens	External	✓	-
AntiSnipe	Interface			
	checkUser	External	✓	-
	setLaunch	External	✓	-
	setLpPair	External	✓	-
	setProtections	External	✓	-
	removeSniper	External	✓	-
	removeBlacklisted	External	✓	-
	isBlacklisted	External		-
Diferencial	Implementation	IERC20		
	<Constructor>	Public	Payable	-
	<Receive Ether>	External	Payable	-
	transferOwner	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	allowance	External		-
	balanceOf	Public		-
	transfer	Public	✓	-
	approve	Public	✓	-
	_approve	Internal	✓	
	approveContractContingency	Public	✓	onlyOwner

	transferFrom	External	✓	-
	setNewRouter	Public	✓	onlyOwner
	setLpPair	External	✓	onlyOwner
	setInitializer	External	✓	onlyOwner
	isExcludedFromLimits	External		-
	isExcludedFromFees	External		-
	isExcludedFromProtection	External		-
	setExcludedFromLimits	External	✓	onlyOwner
	setExcludedFromFees	Public	✓	onlyOwner
	setExcludedFromProtection	External	✓	onlyOwner
	removeBlacklisted	External	✓	onlyOwner
	isBlacklisted	Public		-
	removeSniper	External	✓	onlyOwner
	setProtectionSettings	External	✓	onlyOwner
	setTaxes	External	✓	onlyOwner
	setWallets	External	✓	onlyOwner
	setRatios	Public	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setMaxWalletSize	External	✓	onlyOwner
	getMaxTX	Public		-
	getMaxWallet	Public		-
	setSwapSettings	External	✓	onlyOwner
	setPriceImpactSwapAmount	External	✓	onlyOwner
	setContractSwapEnabled	External	✓	onlyOwner
	excludePresaleAddresses	External	✓	onlyOwner
	setBurnEnabled	External	✓	onlyOwner
	setBurnLimit	External	✓	onlyOwner
	_hasLimits	Internal		
	_transfer	Internal	✓	
	contractSwap	Internal	✓	lockTheSwap
	_checkLiquidityAdd	Internal	✓	
	getCirculatingSupply	Public		-
	getCirculatingSupplyWithRate	Public		-
	enableTrading	Public	✓	onlyOwner

	sweepContingency	External	✓	onlyOwner
	multiSendTokens	External	✓	onlyOwner
	isExcludedFromReward	Public		-
	setExcludedFromReward	Public	✓	onlyOwner
	tokenFromReflection	Public		-
	finalizeTransfer	Internal	✓	
	takeTaxes	Internal	✓	
	_getRate	Internal		

Contract Flow



Domain Info

Domain Name	diferencialtoken.com
Registry Domain ID	2683615263_DOMAIN_COM-VRSN
Creation Date	2022-03-22T23:17:10Z
Updated Date	2022-05-22T02:17:51Z
Registry Expiry Date	2023-03-22T23:17:10Z
Registrar WHOIS Server	whois.launchpad.com
Registrar URL	LaunchPad.com
Registrar	Launchpad, Inc. (HostGator)
Registrar IANA ID	955

The domain has been created 3 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

The Smart Contract analysis reported one critical severity issue. The contract owner has the authority to set an external contract to determine if the transfer should proceed. The external contract is not verified, so it works as an untrusted source. If the external contract contains vulnerabilities then the token can be converted into a honeypot and prevent users from selling. There is also a limit of max 15% fees.

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